An Exploration of Traffic Signal Control using Multi-agent Market-based Mechanisms

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy by

Jeffery Raphael

May 2018

Contents

Illustrations

List of Figures

List of Tables

Abstract

Traffic congestion is a major issue on many urban road networks around the world. The distributed and stochastic nature of traffic has attracted the multi-agent and market mechanism community to the traffic domain which has resulted in many novel approaches to both traffic control and traffic assignment. However, the real-world application of many market-based traffic control systems remains in question because they require technology that has not yet been developed, e.g., autonomous cars.

This thesis focuses on the usage of market mechanisms for traffic control, more specifically, the application of market principles set forth in market-based multi-robot systems to the traffic domain. Thus, the primary goal of this thesis is the design, implementation and evaluation of a multi-agent market-based traffic control system which does not rely on *vehicle agents* and other major changes to vehicles or transportation infrastructure. Evaluation of the traffic control system is conducted on two grid-based maps using six different traffic scenarios. The traffic scenarios represent various traffic patterns which include changes in traffic intensity and direction. The traffic scenarios are simulated in SUMO, an open source, macro traffic simulator. Additionally, performance is measured using three metrics: travel time, traffic density, and number of stops.

This thesis makes five contributions: (*i*) demonstration of the efficacy of a novel multi-agent market-based traffic control methodology; (*ii*) demonstration of the efficacy of a market-based technique for dynamic coalition formation; (*iii*) analysis of three key traffic control parameters used by SCOOT, a popular urban adaptive traffic control mechanism used in over a dozen countries; (*iv*) development of a Python implementation of SCOOT for use on SUMO and (*v*) a thorough evaluation of the novel market-based mechanisms introduced here, along with SCOOT and a reinforcement-learning traffic controller, over a variety of road traffic conditions. This thesis provides a unique insight into the behaviour of three key traffic control parameters and results show that the novel market-based mechanism has the potential to improve traffic performance in traffic conditions that are less than ideal for SCOOT.

Acknowledgements

First, I would like to thank my loving wife, Grace. Her support has been invaluable these past four years. I could not have finished this thesis if it werent for her encouragement during all the long and difficult nights writing. Every day she reminded me to maintain the *eye of the tiger*.

I would like to thank my primary supervisor, Dr Elizabeth Sklar, for her continuous support of my PhD study and related research, for her patience, motivation, and knowledge. Her guidance has helped me immensely in the writing of this thesis. I thank Dr Sklar not only for her academic support and the many opportunities she has brought my way but also for being a friend and mentor. As a mentor, she has taught me more than I could ever give her credit for here. She has shown me, by her example, the hard work and diligence that it takes to be a good scientist. I would like to thank her for encouraging my research and allowing me to grow as a research scientist. I would also like to thank my second supervisor, Dr Simon Maskell, for his guidance. Dr Maskell was instrumental in setting the stage for my research work. I would like to thank him for the countless discussions and brainstorming sessions, especially during the initial phases of my research.

Besides my supervisors, I would like to thank my advisors, Dr Simon Parsons, Dr Katie Atkinson and Dr Danushka Bollegala, for their insightful comments, encouragement and all the hard questions that helped to drive my research. I would especially like to thank Dr Parsons for sharing his expertise on auctions and for being so dedicated to his role as my advisor; he was always willing to make time to give a second opinion or just provide words of encouragement. And I would like to acknowledge all the other staff and faculty members at the University of Liverpool who have been kind enough to extend their help at various phases of my research.

I would like to thank Richard and David, my office mates, for their sense of humour and enthusiasm for *work*. They made all the long hours at my desk possible. I would also like to thank them for their advice on improving my writing, software and conducting research in general. I have gained so much from them, both on a personal and professional level.

Lastly, I would like to thank my family for all their love and encouragement. A great many thanks to my brothers and sisters for supporting me spiritually throughout writing this thesis.

Chapter 1

Introduction

1.1 Introduction

Karl Benz, who is often credited with inventing the internal combustion engine, began mass production of his Model 3 Benz in the late 19*th* century. Although at the time there were competing forms of power, such as steam and electricity, the combustion engine became the most viable option and eventually usurped its competitors. The motor vehicle which uses the internal combustion engine forever changed transportation. Much faster and more comfortable than the horse, the motor vehicle could travel greater distances in a fraction of the time. It brought people, goods and services closer, changing the social and economic fabric of society. In the early part of the 20*th* century, there were an estimated 1.7 million motor vehicles in the UK [4]. Fast forward a little over a century and the UK now has an estimated 36 million motor vehicles [5]. The rise of the motor vehicle has not been perfect and nor did it come without a cost. Cities around the world are now struggling with high volumes of traffic. The popularity of the motor vehicle has given rise to both economic and environmental issues.

Traffic congestion occurs when the volume of traffic exceeds the capacity of the road infrastructure and causes traffic flow to slow down or come to a complete stop, this can also occur after accidents. While the number of drivers has increased dramatically, our road networks are struggling to keep pace with the growth of vehicles. Building new roads or widening existing roads is not always an option. Despite rising fuel prices, motor vehicles remain the predominant means of getting to work for over 60% of commuters in England and Wales [6]. In London, despite having access to public transportation, over a quarter of Londoners still choose to drive to work [6]. During *rush hours*, traffic volume often reaches levels that severely strain current traffic management systems. The sheer number of vehicles sharing the road network has caused congestion to occur even during off-peak hours. Traffic volume and common work hours are just two of the many factors that can grind traffic to a halt. Additionally, antiquated traffic control systems exacerbate the problem. There are many areas that still employ fixed-time traffic signals that do not adapt to changing traffic conditions. Traffic congestion leads to wasted time (idling in traffic), pollution, and lost productivity.

The transportation network is an invaluable component of a city's economic health. Transportation allows for the free movement of both goods and services. The cost of traffic congestion can be measured both in time and money. According to a report put out by the Centre for Economics and Business Research (CEBR) [7], drivers in London waste around 66.1 hours per year waiting in traffic. In 2011, drivers in Europe's three largest economies (UK, France and Germany) spent 39.2, 40.8 and 39.2 hours per vehicle, respectively, idling [7]. The transportation and distribution of goods costs the UK an additional £3.76 billion (€4.94 billion) because of traffic congestion [7]. This includes direct (fuel cost and lost productivity) and indirect (higher cost of goods and services) aspects of idle time. Indirect costs are incurred when businesses must pass the cost of congestion to its consumers. Other European countries face similar monetary losses. Traffic congestion costs France, Germany, and Spain €5.55 billion, €7.83 billion and €5.5 billion, respectively [7, 8]. The estimated annual cost of congestion in the EU is €111.3 billion [8].

However, traffic congestion is not limited to lost revenue. Scientists measure the amount of particulate matter (particles 2.5μ or smaller) to quantify air quality. Another source of air pollution is nitrogen dioxide. It is a harmful gas that causes acid rain which is toxic to plants and aquatic life. Although there are many sources of air pollution (e.g., commercial, institutional, or agricultural), motor vehicles have been identified as a major contributor of particulate matter and nitrogen dioxide [9, 10]. A report by the Environment Committee of the London Assembly [9] found that diesel vehicles can account for 40% of nitrogen dioxide emissions. London is not alone in its struggle with traffic congestion and its effect on air quality. Other cities, such as Beijing, Los Angeles, and Delhi, have similar problems. Studies of particulate matter (and nitrogen dioxide) have found strong evidence that air pollution is a major health hazard.

There is a great deal of interest in developing systems to reduce traffic congestion. The traffic domain presents a number of fascinating problems because it is distributed and the interactions amongst its components can not be easily modelled. It is this complexity that makes the traffic domain so appealing to computer scientists. Furthermore, traffic congestion is a quality of life issue that effects many major cities around the world. Any effort to improve transportation efficiency would have significant benefits to air quality. This is especially important as it appears there are no safe levels of allowable air pollution where particulate matter ceases to have a negative effect [11]. Although it is important to study new methods of tackling traffic congestion, it is even more important that those methods work within the technological constraints imposed by our transportation infrastructure.

1.2 Statement of the Problem

Market mechanisms are a powerful tool for the allocation of resources, e.g., energy, space or goods, because of their well defined structure and minimal communication and computational costs [12, 13]. For this reason, markets have been employed within multi-agents systems as a framework for coordination; this includes multi-agent traffic management systems. Within the traffic domain, markets have been applied in a number of ways, at the intersection level for traffic control (i.e., determining order of movements) and the city wide level where auctions influence the distribution of traffic. However, the literature on market-based traffic management systems reveals major underlying assumptions about the state of the transportation infrastructure: first, that vehicles are able to communicate with the transportation infrastructure (and in some cases with other vehicles); and second, that it is possible to impose a traffic control policy without traffic signals (e.g., the transportation vehicles are fully autonomous). The problem then is, in the absence of such assumptions, how to construct a market-based traffic management system, specifically for intersection control, which does not need additional communication software or devices placed within vehicles or within the transportation infrastructure. This approach presents two challenges: one, providing information on traffic condition for traffic signal timing adjustments without added communication systems; and two, finding a policy for setting traffic signal states such that over saturation and queue spillback are avoided [14]. Additionally, the policy should maintain the safety of the intersection and ensure that no two conflicting vehicle movements ever occur at the same time.

1.3 Purpose and Scope

The main purpose of this thesis is to demonstrate the efficacy of a market-based traffic control system with the following properties:

- Infrastructure Requirements —the traffic control system should rely on currently available transportation infrastructure (e.g., in ground vehicle detectors, traffic signal devices). More specifically, the traffic control systems does not require vehicles to communicate with the transportation infrastructure or with other vehicles.
- Safety —free from collisions caused by permitting two or more conflicted vehicle movements at the same time.
- Communication Overhead —minimal communication requirements to increase reliability because communication is susceptible to corruptions by noise or nefarious acts (e.g., hackers).
- Starvation Prevention —all incoming roadways should be given a minimum amount of green time. That is, all vehicles will be given an opportunity to cross the intersections within a reasonable amount of time.
- Scalable —the traffic controller system should have the ability to grow with the size of the road network.

Furthermore, finding a policy for setting traffic signal states requires working within the parameter space of traffic signals. Thus, this thesis studies three key traffic control parameters and their effects on traffic performance with respect to the novel market-based approach introduced here. The traffic control parameters studied in this thesis are: *split* which is the proportion of the traffic signal timing for a specific roadway; *cycle*, or cycle length, is the length of time for all vehicle movements to occur; and *offset* which is the difference in the starting time of two adjacent traffic signals. Along with the market-based traffic control systems, three other systems are presented in this thesis as benchmarks: SCOOT (a commercially available traffic control system), a traffic control system that learns a policy for setting traffic signal states (using Reinforcement-learning) and fixed time traffic signals. In addition, several SCOOT variants are implemented to investigate the effects of adjusting different combinations of *split*, *cycle* and *offset* on traffic performance (SCOOT is designed to adjust all three traffic control parameters).

1.4 Research Questions

My approach to market-based traffic control raises a number of research questions regarding mechanism construction, traffic control parameters, and coordination:

I. Mechanism Construction. One of the most important objectives of this thesis is to design and develop a more practical market-based traffic control system. That is, a market-based traffic control system which is free of the technological hurdles that are present in many other market-based traffic control system.

   Research Question 1 *How can a market-based traffic controller function without on-board vehicle software (e.g., vehicle agent) or transportation infrastructure upgrades (e.g., communication devices)?*

II. Traffic Control Parameters. The market-based traffic control systems presented in this thesis utilise traffic signals. However, traffic signals are not one dimensional. There are several traffic control parameters that affect traffic performance, e.g., *split*, *cycle*, *offset*.

   Research Question 2 *How can the use of split, cycle and offset adjustments be used to improve traffic performance?*

   Research Question 3 *How does adjusting split differ from adjusting cycle, offset or a combination of the three traffic control parameters affect the performance of market-based traffic control system?*

III. Coordination. Coordinating the signal timing of multiple intersections to allow smoother travel is a common practice in many traffic control systems. Furthermore, groups of coordinated intersections are often fixed. For example, in SCOOT, the road network is organised into small groups of intersections called *regions*. Additionally, membership within regions is fixed during operation of SCOOT. Dynamic coalition formation would allow intersections to coordinate with one another but only when beneficial to traffic performance, that is, a coalition could dissolve when no longer needed.

Research Question 4 *How can intersections in the proposed market-based traffic control system expand their working boundary through the use of dynamic coalitions?*

1.5 Contributions

This thesis contributes to the knowledge of market usage in traffic control. Furthermore, this thesis expands our understanding of traffic control parameters, *split*, *cycle* and *offset*, and their effects on traffic performance. Thus, this thesis makes five contributions:

I. The efficacy of a multi-agent market-based traffic control system which can be deployed within the technological constraints of current transportation infrastructure, e.g., the lack vehicle-to-infrastructure communications, is demonstrated;

II. Demonstrates the efficacy of a market-based traffic control system with dynamic coalition formation to facilitate intersection coordination;

III. A performance of three key traffic control parameters, *split*, *cycle* and *offset*, which are used by SCOOT. Furthermore, the evaluation of the traffic control parameters is conducted with SCOOT and my market-based mechanism, providing a more detailed analysis of how these traffic control parameters affect traffic performance.

IV. Although SCOOT is used in nearly a dozen countries, the SCOOT application is not freely available. This thesis provides a detailed explanation of the SCOOT method for traffic control and a Python[1] implementation of SCOOT for use on SUMO traffic simulator.

V. An evaluation of my market-based traffic control systems, SCOOT and a reinforcement-learning traffic controller, on two road networks and over a variety of road traffic conditions.

1.6 Summary

The remainder of this thesis is organised as follows:

[1] Python code will be made available via Bitbucket.

- **Chapter 2: Traffic Signal Control** – Describes the basics of traffic control at signalised intersections. This chapter also describes common methods used by transportation departments for traffic control.
- **Chapter 3: Literature Review** – Discusses the background literature on market-based traffic control as well as traffic assignment which often overlaps with traffic control. Additionally, this chapter covers the literature related to reinforcement-learning in traffic control.
- **Chapter 4: Market-Based Traffic Control System** – Explains my approach to market-based traffic control; four market-based traffic control systems are presented: SAT, SATQ, MMDOS and DC2.
- **Chapter 5: Evaluation of Market-Based Systems** – Describes the experiment environment, including a description of the traffic simulator and traffic control systems that are used as benchmarks. A detailed description of SCOOT is included in this chapter.
- **Chapter 6: Results of SAT/Q and MMDOS Experiments** – Presents an analysis of the simulation results for SAT, SATQ and MMDOS.
- **Chapter 7: Results of Traffic Control Parameter Experiments** – Presents an analysis of the simulation results for SCOOT and MMDOS variants which employ different combinations of traffic control parameters.
- **Chapter 8: Results of Dynamic Coalition Formation Experiments** – Presents an analysis of the simulation results for DC2.
- **Chapter 9: Discussion** – Provides a discussion of the results from Chapters 6, 7 and 8.
- **Chapter 10: Conclusion** – Provides a summary of my contributions and a discussion of future work.

1.7 Publications

The following publications have been covered in this thesis:

- [15] Jeffery Raphael, Simon Maskell, and Elizabeth Sklar. From Goods to Traffic: First Steps Toward an Auction-based Traffic Signal Controller. In *13th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*, Cham, Switzerland, 2015. Springer International Publishing.
- [16] Jeffery Raphael, Simon Maskell, and Elizabeth Sklar. An Empirical Investigation of Adaptive Traffic Control Parameters. In *Proceedings of the Workshop on Agents in Traffic and Transportation at IJCAI 2016*. CEUR-WS.org, 2016.

- [17] Jeffery Raphael, Elizabeth I. Sklar, and Simon Maskell. An Intersection-centric Auction-based Traffic Signal Control Framework. In Amparo Alonso-Betanzos, Noelia Sanchez-Marono, Oscar Fontenla-Romero, Gary J. Polhill, Tony Craig, Javier Bajo, and Juan Manuel Corchado, editors, Agent-Based Modeling of Sustainable Behaviors, pages 121–142. Springer International Publishing, 2017.

SAT and SATQ have been presented in [15, 17]. MMDOS and early work on dynamic coalition have been presented in [16].

Chapter 2

Traffic Signal Control

2.1 Introduction

This chapter describes the usage of traffic signals in traffic control, that is, an explanation of how traffic signals manage traffic flow. Moreover, it explains three general approaches to traffic control: fixed-time, actuated and adaptive traffic control systems. Additionally, this chapter defines terms used to explain the inner workings of traffic signals, e.g., the traffic signal parameters that are adjusted to improve traffic flow.

Traffic control is the process of managing traffic flow with traffic signals. However, traffic flow may become compromised and slow to a crawl with improperly timed traffic signals. Traffic congestion is the point where the number of vehicles utilising the road network exceeds its capacity. That is, all road networks have an upper bound which limits the total number of vehicles it can handle. Additionally, other events, such as road repairs and accidents can diminish the capacity of the roadway. The upper bound is set by a number of factors such as availability of parking, lane width, road geometry (this includes turning lanes), and traffic rules and regulations. It is the last factor, traffic rules and regulations, that this section is focused on. More specifically, the use of traffic signals to manage traffic flow. As such, my work revolves around interrupted traffic flow as opposed to traffic flow found on motorways (highways). In other words, traffic flow that is interrupted by traffic signals [18, 19].

Unlike signalised roadways, congestion on motorways is mainly caused by the interplay between traffic speed, volume (number of vehicles passing a point in an hour) and density (number of vehicles per unit of road, e.g., per kilometre) [1]. More specifically, as traffic density increases, a critical point is reached where both traffic speed and volume begins to decrease [1]. Additionally, if traffic density continues to increase then a traffic jam occurs where traffic comes to a complete halt. The relationship between traffic speed, volume and density underscores the importance of maintaining a balance between traffic demand and road capacity. This is where the two types of road networks, motorways and signalised intersections, have common ground. On both motorways and signalised intersections (roadways), demand cannot exceed capacity if traffic is to flow freely. Furthermore, traffic signals can negatively impact capacity (a scenario which is not possible on motorways).

Traffic signals employ various lights (and sometimes coloured arrows) of different durations to direct traffic at an intersection. Traffic signals are one of the most popular means of managing traffic flow because of their simplicity both from a driver's and traffic engineer's perspective. Thus, traffic signal timing, i.e., the duration of each light sequence, plays an important role in traffic efficiency. Poorly timed traffic signals can increase air and noise pollution, travel times and even cause accidents. Manipulating traffic signal timing presents an opportunity to improve traffic flow without costly changes to the transportation infrastructure. For traffic engineers who seek to mitigate the causes traffic congestion optimisation of traffic signal timings are a far more economical option than structural changes to the road network, e.g., building new roadways.

My work is about maximising current usage of road networks within the limits imposed by the other factors listed above. The optimisation of traffic signals is not limited to isolated intersections and instead spans entire cities. The problem of traffic control stands to benefit from improvements in vehicle telemetry, information technology and (wireless and or mobile) communications.

The traffic signal control problem is one of timing, regardless of whether or not this is at an isolated intersection or a large network of intersections. At an isolated intersection, the concern is more or less one of throughput, i.e., the number of vehicles the intersection services during a pre-defined period of time. However, when the problem encompasses a much larger area, with multiple intersections, optimising signal timings becomes a great deal more difficult. Traffic conditions at one intersection influence traffic at other intersections. For example, improving signal timings at one intersection may actually cause traffic delays at adjacent intersections. More importantly, the farther away two intersections are from one another, the more difficult it is to model the relationship between the two intersections. Additionally, the underlying process for this degradation is that vehicles are driven by people, each with their own beliefs, desires and goals.

In general, traffic control strategies can best be described as part of a closed control loop [20]. Papageorgiou *et al.* [20] describes the basic components of the traffic control loop which consist of two components that dictate traffic flow behaviour: *control inputs* and *disturbances*. Disturbances represent traffic demand and other traffic incidents that are measurable (e.g., traffic incidents that can be detected by vehicle detectors) as well as predicted traffic events. The analysis of the traffic conditions (or disturbances) results in adjustments to control inputs. Traffic control systems vary in the source of traffic information, e.g., vehicle detectors versus cameras or GPS, and the control inputs they are designed to optimise.

Traffic signals are one of many tools used by traffic managers to regulate the flow of vehicles. However, it is also one of the most important in congestion control. In general, the goal of traffic managers is to create synergy between those that use the road network and those impacted by its presence (e.g., pedestrians and cyclist). Hence, there are many factors (social, economic and environmental) that influence the management of road networks. Some of these factors range from safety and accessibility to parking and promoting cycling.

This dissertation uses signalised intersections because the policy is easily understood and reproduced in simulation. The inner workings of traffic signals are universal; that is a traffic signal in the U.S. works on the same principle as a traffic signal in U.K. Although there are other forms of intersection control polices, their behaviours (i.e., management of vehicle movements) are more prone to corruption and thus the intersections' behaviours are more difficult to replicate in simulation. For example, the performance of stop and yield signs is more heavily dependent on human behaviour than traffic signals. That is, humans do not always come to a full stop at an intersection or when there are multiple vehicles at an intersection, follow the right-of-way rule.

Traffic signals are the most popular form of intersection control policy. They offer many advantages over other control policies (e.g., stop sign or traffic police). One of the most important aspects of traffic signals is that they make intersections safer. Traffic signals can reduce certain types vehicle accidents [18] and provide a means to blend pedestrian traffic with vehicle traffic.

When used correctly, i.e., with the right intersection and signal timings, traffic signals can minimise right-angle, turning, and pedestrian accidents [18]. On the same token, traffic signals can increase rear-end collision. That is, a light change may force a driver to make a sudden stop which in turn gives the following vehicle insufficient time and distance to break safely. Improperly timed traffic signals can cause excessive amounts of delay because signal timings are not tuned for the level of traffic. Furthermore, if traffic signals take too long drivers may assume the traffic signal is broken, especially if there is little cross traffic, and simply revert to their own judgement as to when to proceed or not. Traffic signals also allow for pre-planned sequences of vehicle movements through an intersection. Traffic signals, assuming they are well timed, can increase the general capacity and efficiency of an intersection. Groups of traffic signals can be coordinated to allow entire platoons (a group of vehicles in close proximity to one another, travelling at a similar speed) to traverse several intersections at speed. However, the main purpose of installing traffic signals are to increase capacity and safety of intersections [18].

The reason why traffic efficiency is not the main goal of installing traffic signals is that it is more difficult to produce optimal signal timings than simply using fixed-times. More importantly, poorly timed signals actually increase accident rates and greatly increase delay [18]. Hence, many traffic signals use fixed-timings that are rarely changed. Traffic engineers look for several intersection attributes when deciding whether or not to install a traffic signal. Traffic and pedestrian volume, speed limits and the road/intersection geometry, accident rates and even pedestrian demographics (e.g., age) are all considered during the decision making process [18].

One of the important uses of traffic signals is to ensure the safe use of intersections.

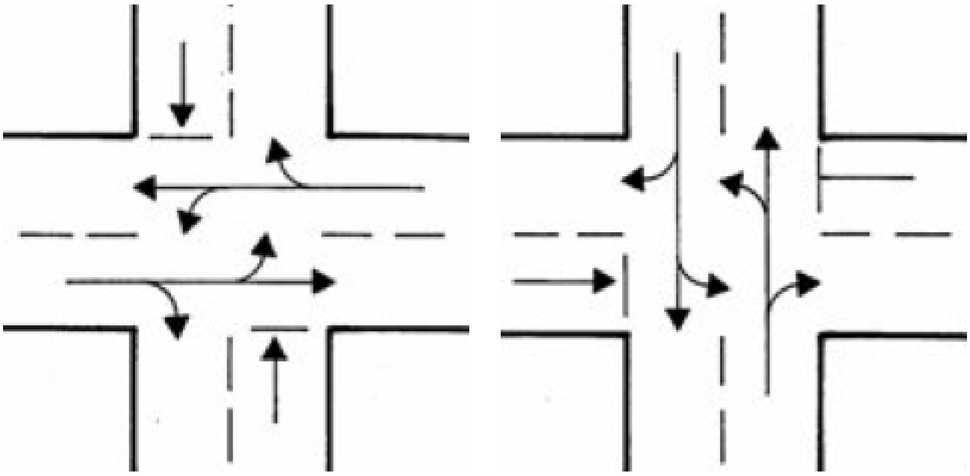
More specifically, the traffic signals are used to resolve conflicts between vehicle movements, e.g., through traffic in one direction and a right turn from a perpendicular street. However, in doing so, traffic signals become a significant source of delay. This added delay is one of the reasons doing away with traffic signals is the holy-grail of traffic control. The delay caused by traffic signals has several sources tied to how they function, thus, there are different types of delays. Roess *et al.* [18] describe two forms of delay: *approach* and *time-in-queue* delay.

The approach delay is the combination of the delay caused by two events at signalised intersections. First, as a vehicle approaches a red light, it must come to a full stop. Second, when the traffic signal turns green, the vehicle has to accelerate. Both of these events causes delay, i.e., the vehicle loses time not travelling at speed, and together form the approach delay. Time-in-queue is the total time the vehicle spends in a queue including the moment it has begun to accelerate but has not crossed the stop line.

Vehicle delay at a signalised intersection can be explained in terms of the rate at which vehicles arrive (*arrivals*) and depart (*departures*) an intersection [18]. Thus, there are three possibilities, two of which lead to an increase in delay. First, signal timing is well tuned for prevailing traffic demand; second, some portions of the signal timing fail to meet demand and finally, the intersection is functioning at or above peak capacity. Under normal operating conditions during green lights, *arrivals* equals *departures*. In other words, the *green* time (amount of time allotted to a set of vehicle movements) is sufficient to purge the entire vehicle queue every time. The *cycle length* plays a critical role in the ability of an intersection to manage its queues, after all, it is during the *red* time that queues form. The second possibility is that some *cycles* may experience arrival rates that exceed the departure rate, i.e., *arrivals* are greater than *departures*, in which case a queue begins to form that remains even after a *cycle* has completed. Lastly, the third possibility occurs when the signal timings are either poorly designed or or a traffic event has occurred which it can not manage. In cases where *arrivals* exceeds the intersection's capacity (the maximum number of vehicles the intersection's geometry and signal timings can manage), queues form and continually grow on every *cycle*. The formation of queues and the resulting delay highlights the importance of cycle length and green time in minimising intersection delay. Allocating appropriate amounts of *green* time is the most effective means of controlling delay [18, 21]. However, there has to be balance between the *green* time of competing vehicle manoeuvres.

Although traffic signal timing adjustments are primarily driven by the desire to reduce travel delay (or increase network throughput), there are a few intersection conditions that must be avoided:

- Green times that are too short. Insufficient green times do not give drivers enough braking distance and may increase rear-end collisions.
- The opposite, long green times, are problematic as well. Excessive green times cause longer queues to form on the links that are waiting for their phase.



(a) Phase 1 (b) Phase 2

FIGURE 2.1: Allowable vehicle movements in a two-phase plan. In the first phase, traffic heading East/West is permitted to use the intersection while in the second phase North/South bound traffic allowed to use the intersection [1]

.

- Red times that are too short can trap pedestrians in the middle of a busy roadway.
- Lastly, red times that are too long can cause *starvation* (one or more incoming links that have to wait long periods for their green phase or the green phase is too short).

2.2 Phase Plan

Traffic signals manage conflicting movements at an intersection by allowing and restricting movements during set time periods. The traffic signal timing is based on a *phase plan*. A *phase plan* describes the sequence of lights a traffic signal will emit. Developing a *phase plan* involves identifying the different sets of safe movements. The most basic *phase plan* is the two-phase plan, shown in Figure 2.1. Assuming an intersection has two intersecting streets, the two-phase plan would assign a *phase* to each street.

Figure 2.1 is an illustration of an intersection serviced with a two-phase plan. This particular intersection has four incoming links, each a single lane. Also, there is not a dedicated turning lane, thus, any vehicle that has to make a turn must wait until it is safe to do so. While one phase is in play, all others phases must show red. That is, only one phase is active (green) at any given moment and therefore only a subset of movements are allowed during that same moment. The time required for all the phases to pass is the *cycle* length.

A *phase*, illustrated in Figure 2.2, is a portion of a traffic signal timing that is given to a set of vehicle movements [22]. A *phase* includes a green *interval*, followed by an amber and then a red *interval* (and any transitional lights) that are assigned to a movement or a set of movements. Each interval serves a different purpose:

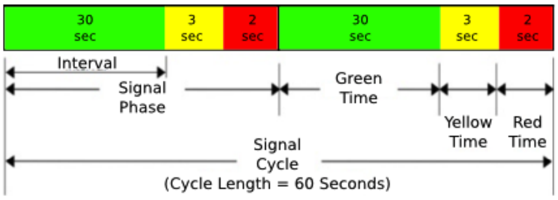


FIGURE 2.2: Sample traffic signal timing.

- green —during this period subsets of intersection movements are allowed. Often, there is a minimum amount of green time specified for a phase. The green *interval* is also referred to as the *split*.
- amber —this period is used to warn drivers that the light is about to turn red. It is used to provide drivers with enough breaking distance, i.e., the ability to safely and comfortably come to a full stop.
- red *clearance* —this is the amount of time between the ending of one phase and starting of another. It is issued to ensure vehicles that have entered the intersection as the signal turned red have enough time to clear the intersection space.

Additionally, in the UK, traffic signals will show red and amber prior to turning green; this allows drivers to prepare to go. Together, all the *phases* form the signal timing for a traffic signal (or a complete cycle). The traffic signal cycle is a collections of phases that service all the incoming links in an intersection.

Each *phase* in a *phase plan* represents a set of movements and depending on the intersection there may be far more than two *phases* needed to control the traffic signal. Dividing all possible legal movements into subsets is normally done by traffic engineers within the guidelines set by their local transportation officials (e.g., traffic engineers in the U.S. use MUTCD [18]).

2.3 Modes of Operation

Probably one of the more popular methods of operating traffic signals is as fixed-time controllers. With fixed-times, traffic signals repeat the same sequence of phases and the length of each phase remains the same for every *cycle*. The process of developing fixed signal timings is actually more involved than one would expect. First, field measurements have to be collected (i.e., congestion periods, vehicle delay and queue lengths). Some of the data may require engineers to visit the site (intersection) of the traffic signal and/or special equipment. Second, the data has to be analysed and searched for exploitable traffic flow patterns. Third, a preliminary signal timing is produced based on the results from the data analysis. Lastly, the signal timing is installed in the field. However, more data has to be collected and the timings fine-tuned while deployed. The entire process can be repeated any number of times and require months of work —depending on the goals of the engineers. The benefit of fixed-time traffic signals is that they are inexpensive and easy to maintain. Fixed-time traffic signals are ideally used where traffic flow is fairly constant, i.e., the ebb and flow of traffic is highly predictable. The major disadvantage of fixed-time traffic signals is that it only works if demand is stable. In reality traffic flow is far from stable, instead there can be sudden, and more importantly unpredictable, changes in demand. Pre-time or fixed-timed traffic signals offer a single response to traffic flow and this can easily result in the failure of some cycles or the entire signal timing.

Actuated traffic signals (ATS) have a control logic which allows them to process information about current traffic conditions and then respond accordingly [23]. ATS are completely reactive; adjustments to signal timing are made based on current traffic demands. Additionally, signal timings are selected from a small set of possible timings [23]. Actuated traffic signals are a slightly better solution than fixed-time traffic signals. There are two main type of ATS: *semi-actuated* and *full actuated* [18]. Intersections that utilise semi-actuated traffic signals have vehicle detectors on the links that have the least traffic volume (there are no vehicle detectors on the major links). The way semi-actuated traffic signals work is the major link always has the green light until the traffic signal is alerted (via the vehicle detector) of traffic on the minor link. Minor links are given a maximum allowable amount of *green* time before the signal reverts back to the major link. The traffic signal may also revert back to the major link if it detects that there are no additional vehicles on the minor link. The primary purpose of semi-actuated traffic signals is to provide a safe suspension of traffic on a major artery where there is little cross traffic. Fully actuated traffic signals on the other hand, utilise vehicle detectors on all incoming links. They follow pre-programmed rules to determine which phase to operate given current traffic conditions. Under fully actuated traffic signals, the *split*, *cycle* and phase sequence changes from one *cycle* to another.

ATS can be designed to have variable phase sequence, *green* time or cycle length [18]. ATS operate well on isolated intersections, where there is no need for the intersection to coordinate its signal timing with neighbouring intersections. The intersections must share the same cycle length to optimise traffic flow through multiple intersections which would not always be possible with ATS. Additionally, ATS are more costly than fixed-time traffic signals because ATS require sensors to detect vehicles and/or pedestrians.

2.4 Adaptive Urban Traffic Controllers

Adaptive urban traffic controllers (UTC) fine tune traffic signal timings in real-time to manage traffic demand over a large area. UTCs utilise various road sensors to collect information on traffic conditions and then update the *split* times and cycle lengths of the traffic signals under their control. Historically, macroscopic optimization programs, which are primarily deterministic in structure, such as TRANSYT-7F and SYNCHRO have optimized traffic signal timing plans [24]. Some of the first UTCs, such as TRANSYT-7F, utilised fixed-time traffic signals; the signal timings were optimised offline using historical data. Later, more advanced UTCs came about that selected an optimal signal plan depending on current traffic conditions, e.g., SCAT (Sydney Coordinated Adaptive Traffic System); many UTCs in the US work in this manner [25]. However, such systems have a limited number of signal plans to choose from. The major differences between UTCs can be found in the technology used to monitor traffic, the signal timing optimisation methods employed, the traffic control parameter that is adjusted (e.g., *split*, *cycle* length, *offset* or phase order), and frequency of parameter adjustments. Examples of UTCs are SCOOT (Split Cycle Offset Optimisation) [26], RHODES (Real-Time Hierarchical Optimised Distributed Effective System) [27], and OPAC (Optimisation Policies for Adaptive Control) [28, 29]. Some UTCs are also able to prioritise public transportation and emergency vehicles and even make adjustments for traffic accidents. The main difference between UTCs and actuated traffic signals is that there is more decision making in the UTCs, as opposed to the ATCs. UTCs analyse current traffic conditions and evaluate possible actions that may improve traffic performance, unlike, actuated traffic signals which are more or less reactive and work with a very small set of actions (or plans) that remain fixed.

One of the most important jobs of adaptive urban controllers is to coordinate traffic flow across multiple intersections. *Green wave* formations are probably the most common form of traffic signal coordination. Green waves occur when a series of traffic signals are specifically timed so that a platoon of vehicles leaving one intersection will reach another (downstream) intersection just as the intersection turns green. The result of a green wave is a group of vehicles passing through an artery with very few stops because of red lights. In other words, the coordinated traffic signals form a linear path where vehicles do not have to turn to reach the next intersection in the green wave. Traffic engineers adjust signal timings to target platoons of a certain size and speed. Green waves, when properly tuned, can greatly reduce vehicle emissions, noise and long travel times caused by signalised intersections.

Green waves are implemented by offsetting the starting time of green lights between two adjacent intersections. An *offset* is the difference in the starting time of the green interval between two adjacent intersections. That is, if an intersection's green light occurs at time t, then the green light of the intersection downstream will begin at time $t + z$ where z is the *offset*. Thus, the most cutting-edge UTCs dynamical build signal plans on the fly, i.e., *splits*, *offsets* and *cycle* lengths are changing from one *cycle* to another. All the traffic signals that form the green wave must have the same *cycle* length [18]. If any of the traffic signals function with a different cycle length, then the offset (between intersects) can not be controlled.

Figure 2.3 illustrates the formation of a green wave along four intersections. Vehicle(s) leaves the first intersection (labeled reference signal) and reaches each subsequent intersection while the signal is showing *green*. Traffic engineers calculate the ideal offset between two intersections given the distance between the intersections and the average

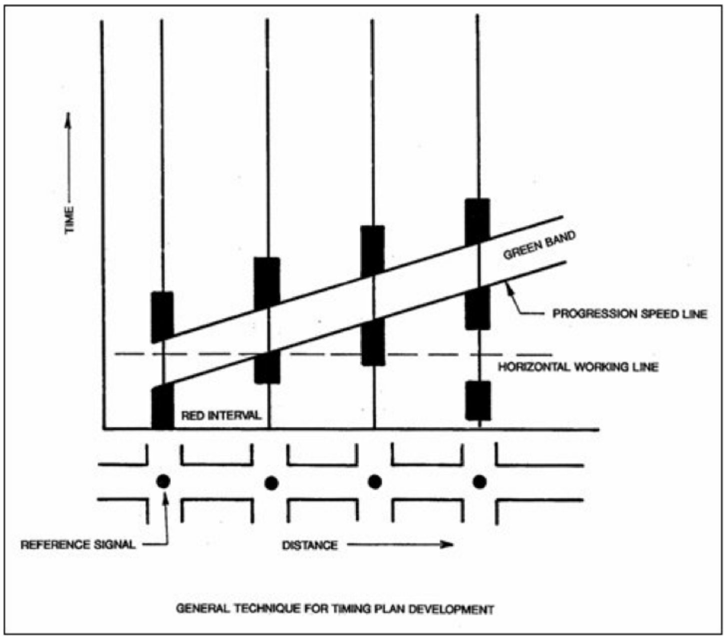


FIGURE 2.3: Illustration of green wave formation. Traffic signals are timed to allow platoons to traverse multiple intersection with minimal stops. The green band represents a window in time at each successive intersection (starting at the reference signal) where vehicle(s) will pass through the intersection without having to stop [1].

platoon speed [18], finding the latter is the more difficult of the two terms that are needed. The average platoon speed has to be an accurate measurement of a typical platoon, i.e., across weeks or months of traffic. Additionally, many factors affect the average platoon speed, such as weather, time of day and the types of vehicles that make up the platoon. The difficultly level of finding an offset varies with road geometry, e.g., single versus multi-lane roads, two-way versus one-way and two-phase signals versus multi-phase signals. The interplay between *offsets* in one direction and the *offsets* in the other direction presents a much more difficult offset optimisation problem than on one-way roads. Poorly timed offsets can have a major effect on delay as large numbers of vehicles can become trapped at an intersection and must wait for the next *green* light to proceed.

2.5 Summary

Traffic signals rely on a quite simple schema to safely instruct drivers on which vehicle movements are allowed at any given moment. Although safety concerns are a large component of traffic signal usage, traffic signals are also a key source of delay. Poorly timed traffic signals can significantly slow traffic, causing an increase in travel times and stops. Traffic signals are employed in several ways, as fixed-time, actuated systems, and the more advanced, adaptive traffic control systems. Moreover, this chapter explains the three traffic control parameters: *split*, *cycle* and *offset*, that are at the centre of many traffic control systems. The next Chapter discusses the background literature on traffic control and traffic assignment.

Chapter 3

Literature Review

3.1 Introduction

Traffic management systems can benefit greatly from AI and many other subfields of computer science. Structural changes to road networks are far more problematic than upgrading the systems that manage traffic flow. For example, widening roadways, building overpasses or installing roundabouts are just a few of the possible options for tackling congestion. However, these options may be too costly or impractical in urban areas. Furthermore, building new roads may actually hurt traffic performance [30]. Computer science promises new ways of improving traffic flow without incurring the construction costs of modifying the transportation infrastructure. Traffic management is a broad term that encompasses a number of measures used to ensure the safe and efficient use of road networks. This includes setting speed limit, scheduling traffic signals, installing islands, movement and parking restrictions and establishing dedicated lanes (e.g., bus lanes or multi-occupancy lanes). The traffic domain can be broken into two sub-domains: traffic assignment and traffic control. Regardless of which is tackled, the development of better traffic management systems stands to reduce the amount of time lost idling in traffic, fuel usage, and vehicle emissions.

Traffic control refers to the use of traffic signals to regulate the movement of vehicles through an intersection. In general, the traffic control problem can be stated as that of finding a policy for setting traffic signal states such that traffic flow improves while the safety of an intersection is maintained and conflicts amongst movements are resolved (including pedestrian and cyclist movements) [18]. Such a policy could take into consideration only traffic conditions at the intersection but can also incorporate information from neighbouring intersections. The policy should determine which movement(s) are allowed at any given moment in time. An optimal policy could minimise travel time or it could attempt to optimise other aspects of traffic, such as number of stops per vehicle or *queueing* time (i.e., how long a vehicle waits at a particular intersection before it is able to pass). Finding the optimal traffic signal timing is a non-trivial operation for a number of reasons. Traffic is composed of many elements, e.g., vehicles, pedestrians, cyclists and traffic control devices. Furthermore, traffic is geographically distributed and the interactions amongst its components are highly complex [31]. Traffic signal timings function under rigid temporal constraints which may be represented as discrete variables. Therefore, traffic control behaves in many regards like a combinatorial optimisation problem (e.g., TSP) [20]. Scale is also an issue with traffic control. Any reasonably sized road network will have dozens of intersections, compounding the problem of finding an optimal traffic signal timing [20, 32, 33]. Adaptive traffic control systems that work in real time must also find a solution within a very small time window in order to function properly.

While this thesis focuses on traffic control, this chapter includes systems that tackle traffic assignment as well. Traffic assignment deals with the distribution (route selection and guidance) of vehicles in a road network. The aim of traffic assignment is to guide driver route selections so as to prevent the occurrence of congestion. The methods used to address traffic assignment often overlaps with traffic control. The remainder of this chapter is organised into the following sections: Multi-agent Systems and Traffic (Section 3.2), State-of-the-Art Traffic Systems (Section 3.3), Markets (Section 3.4), Market-Based Traffic Management (Section 3.5) and Reinforcement Learning in Traffic Control (Section 3.6).

3.2 MAS & Traffic

Multi-agent systems (MAS) is a field of computer science that deals with the construction of complex systems composed of autonomous agents. In MAS, agents represent real world entities that act autonomously and in some cases intelligently [34]. Agents can be reactive, adaptive, or even have the ability to learn from their experiences [34]. Agents can also be designed to have social behaviours. The multi-agent paradigm presents new principles for abstracting problems (and solutions) such as planning, cooperation, collaboration or negotiations [35]. Yet, it also draws upon other fields such as machine learning and economics [36]. MAS has found applications in manufacturing, real-time control systems and electronic commerce [31] as well as computer related fields such as gaming, simulation, and computer security [35]. MAS has also been applied to various areas of transportation such as air [37] and rail transport [38] and fleet management [39].

Computer scientists have investigated many ways to cope with the complex and dynamic nature of traffic. The multi-agent systems paradigm is ideal for complex systems that are dynamic and distributed [31, 38]. The different elements of traffic can be viewed as a large collection of autonomous agents. Multi-agent systems provide capabilities that are excellent for traffic control [40]:

- efficiency —agents act in parallel;
- robustness —there is no central point of failure; and
- scalability —adding new agents does not always incur a computational cost.

Modern approaches to traffic control systems rely on mathematical models (of traffic) and/or optimisation algorithms, e.g., OPAC, SCOOT, SCAT and RHODES [36]. However, these methods are difficult to operate and maintain and do not scale well [36]. MAS offers a more flexible approach for traffic management and control systems [36]. Modelling traffic as a large collection of *autonomous agents* allows us to apply a wide range of methodologies for defining the relationships between the different elements of traffic.

Furthermore, agent systems require a mechanism for decision-making [41]. Interactions amongst agents can take many forms depending on the system architecture and the sub-field employed, e.g., machine learning, market mechanisms or genetic algorithms. Agent autonomy and their interactions form a potent combination. For example, traffic signals, signs and/or cameras can be represented as agents and given more autonomy and thus increase adaptability [42]. Hence, the MAS literature is filled with various approaches that apply many AI techniques such as Reinforcement-Learning [19, 43], Fuzzy logic [23, 44], Artificial Neural Networks [45, 46], Genetic Algorithms [47, 48].

3.3 State-of-the-Art Traffic Systems

Autonomous cars are poised to revolutionise transportation. Autonomous cars are vehicles that utilise input from sensors (e.g., radar and lasers) to drive themselves. There are many advantages of having a transportation network of fully autonomous cars. The most important is the removal of humans from the control loop. Human drivers get distracted, suffer fatigue or simply drive recklessly exposing themselves and others to great danger; without human drivers, accidents caused by human error would be drastically reduced. There were nearly $25,700$ road fatalities in the EU in 2014 [49]. The US had nearly $32,719$ fatalities due to traffic accidents for that same year [50]. Autonomous cars will not only improve transportation efficiency and safety, but also will change commuting. On average, a single London commuter wastes 66.1 hours a year in London traffic [7]. With autonomous cars, those hours can be spent more productively. A transportation system consisting of only autonomous vehicles is considered the holy grail by many.

Dresner and Stone [51] propose a reservation-based traffic control systems which utilises autonomous cars. Dresner and Stone removed the two elements of traffic that are most responsible for delay and accidents: human drivers and traffic signals. In the reservation-based traffic control systems, vehicles request reservations to use the intersection. The reservation system has two key agents: *intersection agents* and *vehicle agents* (the autonomous car). The vehicle agents use a communication protocol to gain a window in time and space which allows for safe passage through the intersection. Thus, vehicles have vehicle-to-infrastructure ($V2I$) capabilities as well as being autonomous. The vehicle agent also ensures that the vehicle complies with the reservation, i.e., the vehicle agent maintains the correct velocity to fulfill the reservation. In [51], reservations are allotted on a first-come, first-served bases. In other words, as a vehicle nears the intersection, it sends telemetry data (e.g., estimated time of arrival and maximum velocity) to the intersection agent. The intersection agent uses that information to project the time and space in which the vehicle will occupy the intersection space. If the intersection manager determines the vehicle's trajectory is safe, then it sends a message back to the vehicle accepting the reservation. However, if the intersection can not safely accommodate the vehicle, it responds with a rejection message. Upon receiving the rejection, the vehicle slows down and tries to attain a new reservation. It is the intersection's responsibility to guarantee the safety of reservations, i.e., no two reservations are in conflict.

Dresner and Stone [51] used two metrics to measure performance: throughput and delay. Dresner and Stone [51] compared their reservation-based approach to two other traffic control schemes: overpass and traffic signals. Overpass simulates a road network with absolutely no traffic signals or conflicting vehicle movements at intersections, i.e., at intersection vehicles no longer need to stop or yield to cross traffic. Traffic signals represent traditional traffic signals which utilise fixed signal timings. Dresner and Stone's [51] results showed that their reservation-based system drastically reduced intersection delay. Delay times with the reservation-based system were nearly the same as installing an *overpass*.

Dresner and Stone's [51] initial reservation-based systems had some limitations, e.g., vehicle velocity (speed and direction) was restricted and the communication protocol and intersection control policy were too intertwined. Dresner and Stone improved their reservation-based system in [52, 53]. Restrictions on vehicle behaviours were removed, vehicles could make turns and accelerate even while in the intersection. Dresner et al. [52] also decoupled the communication protocol from the control policy. This allowed the reservation system to mimic traditional traffic control devices, e.g., stop signs and traffic signals. Additionally, the protocol was adapted to prioritise emergency vehicles.

As a multi-agent based system, there is room in Dresner and Stone's reservation-based system to apply other sub-fields of computer science to the allocation of reservations. Dresner *et al.* [54] describes the potential for AI techniques in its reservation-based system. For example, intersection managers can use machine learning to optimise the assignment of reservations. In the case of prioritised vehicles (e.g., ambulances), an intersection can learn the relationship between priority and reservation times to minimise wait or delay of those vehicles with high priority [54]. Also, it is possible to utilise a market mechanism within the the reservation system which would allow driver agents to bid on reservations [54]. In a market-based reservation system, driver agents can learn bidding strategies and/or intersection managers can learn how to price reservations. Also, with additional information from the intersection manager, driver agents can learn optimal changing policies or better estimates of arrival time (i.e., time to intersection space) [54].

Dresner and Stone's original work with autonomous vehicles [51–53] was evaluated on a single intersection. Hausknecht *et al.* [55] investigated the performance of AIM (Autonomous Intersection Management, Dresner and Stone's reservation-based system [53]) on a small road network. Hausknecht *et al.* [55] also went a step further by testing different navigation policies, thus, both traffic control and traffic assignment are addressed in [55]. The navigation system represents the final step in multi-agent based traffic management systems. Once vehicles become intelligent agents, that intelligence is extended from simple control to navigation or route selection. Hausknecht *et al.* [55] demonstrated in simulation that such a traffic management solution could increase vehicle throughput while decreasing travel time.

Hausknecht *et al.* [55] utilised AIM [53] to explore navigation policies in a multi-intersection environment and demonstrated that internal traffic information from AIM can be used for path planning, i.e., the autonomous vehicles make decisions about the path that they will traverse in the network in hopes of reducing travel time. This is the same information that is used in AIM to allocate reservations for intersection use, e.g., vehicle arrival time, velocity and acceleration. In [55], vehicle agents have a *navigation policy* which determines the path the vehicle will take to its destination. Hence, vehicle agents in [55] are required to notify an intersection that it, the vehicle, will in the near future request a reservation. This particular step is not present in the original AIM where vehicle agents simply requested a reservation when needed without prior notification. Vehicle agents also send the intersection positional data, e.g., which incoming link the vehicle is currently on. *Intersection Managers* use this information to calculate the *expected traversal time* (the estimated amount of time vehicles spend on the incoming link including the time to cross the intersection). Furthermore, the *intersection managers* broadcast the *expected traversal time*, i.e., this information is available to all vehicle agents. Vehicle agents perform an A* search to find the path with the shortest total *expected traversal time*. Hausknecht *et al.* evaluated five navigation policies on a 2×2 grid road network and found that when vehicles utilised the A* with *expected traversal times* for path finding, the delay was significantly reduced compared with vehicles that used the distance based A* path finding algorithm.

Of course, Dresner and Stone are not alone in the pursuit of traffic control systems that take advantage of autonomous vehicles. Abdelhameed *et al.* [56] developed a multi-agent based traffic control system with vehicle agents (for vehicle control) and intersection agents which employ a trajectory-based intersection control policy. In [56], the intersection agent or Intersection Manager Agent (IMA) is responsible for controlling movement through the intersection. The IMA continually communicates with the vehicles within its range, i.e., those vehicles that are about to traverse the intersection associated with that particular IMA. The IMA receives information from vehicles and predicts the path the vehicle will take through the intersection. These predictions are used to determine if a trajectory is safe; and if the trajectory is deemed unsafe, the IMA will send commands to the vehicle. Commands from the IMA are vectors describing an amount of acceleration (or deceleration) needed to avoid a collision. After receiving a command it is then the responsibility of the vehicle agent to ensure that the vehicle reaches the appropriate velocity. Abdelhameed *et al.* [56] showed a decrease of 62% in travel times.

A transportation system complete with autonomous cars is an alluring concept. After all, autonomous cars do not get tired, distracted or drunk [53]. Autonomous cars have the potential to put an end to many of the dangers involved in having human drivers. However, wide-spread deployment of autonomous vehicles in real-world environments is not a near-term reality. There are many challenges that remain before self-driving cars will be used by the masses. The transition from human-driven cars to a fully autonomous transportation network is a huge operation.

The development and deployment of autonomous cars is on-going. Google's self-driving cars are widely talked about, with a fleet of autonomous cars that have collectively covered over $700K$ miles [57]. Yet, these cars navigate using special maps that have enhanced information, such as location of traffic signals and driveways. As well, they cannot avoid unmarked potholes and would not be able to obey commands from a traffic officer [57]. It is estimated that by 2040, self-driving cars will completely take over human-driven cars [58, 59]. The full scope of adopting self-driven cars is enormous, involving not just economic issues but social and ethical ones as well. Advancements in self-driving cars are starting to appear in current car models in the form of *semi* autonomy. For example, Tesla Motors has *Autopilot*. *Autopilot* uses radar, cameras, GPS and ultrasonic sensors to steer the car in order to stay within a lane. It can even change lanes when told to do so. There are also vehicles that have automatic cruise control, automatic parking and automatic braking features. The recent death of a driver using Autopilot [60] has raised some interesting questions concerning the use of Autopilot, mainly, *who exactly is responsible for accidents involving autonomous (or semi-autonomous) cars? are systems such as Autopilot (considered a precursor to fully autonomous cars) up to the task of having full control.*

Another issue is the current state of connectivity. The communication infrastructure necessary for broad vehicle-to-vehicle ($V2V$) and $V2I$ currently does not exist. In the USA, the National Highway Traffic Safety Administration (NHTSA) is currently pushing for the use of $V2V$ technology nationwide, arguing that it could dramatically reduce accidents by warning of dangers ahead; but, to date, there is no nationwide agreement or timeline for implementation. Car manufacturers have yet to agree on international communication protocols. Even the physical specifications for the communication devices have not yet been devised.

The lagging development of self-driving and connected cars is problematic for traffic control systems that rely on vehicle agents. We are decades away from having connected self-driving cars and the necessary transportation infrastructure. Assuming society as a whole is fully committed to achieving this goal, for many years there will be a mix of traffic on our roads. That is, traffic will be composed of human drivers, semi-autonomous vehicles and self-driven cars. Dresner and Stone have foreseen this future and added provisions for this in their reservation-based system [53]. Although this suggest that such a system can be progressively implemented, the cost of replacing traffic signals and installing communication devices would be prohibitive. There is also the possibility that development stalls due to technological or social hurdles. Traffic congestion remains a pressing issue in many urban areas around the world. Thus, there remains a need for traffic control systems that are designed to function with currently available transportation infrastructure.

The work for this thesis is motivated in part by the reality that universal deployment of autonomous vehicles is too far off into the future to ignore currently available transportation technology. Given the most likely time frame for the development of a transportation system based solely on fully autonomous vehicles, more practical and feasible solutions should be explored considering the problem of traffic congestion in urban environments. Thus, in my work, I propose using existing transportation devices that are widely available, e.g., in-ground vehicle detectors. Furthermore, my approach does not require vehicle agents, $V2V$ or $V2I$ capabilities.

3.4 Markets

Market mechanisms can be employed as a framework for resource allocation and inter-agent interactions in multi-agent environments [13]. Resources can be time, space, power, or more tangible items such as memory or communication media [13]. Markets provide a scheme for agent coordination and in turn allow agents to better function in environments with limited resources or dynamic events [61]. Markets are a common means of achieving cooperative behaviours in distributed systems such as multi-robot systems [35, 61]. Auctions are probably the most popular market mechanism; an auction is a protocol for the allocation of resources or goods using a *pricing* system [13]. Parsons *et al.* [62] describes three operations in auctions that are commonly found in many market-based multi-agent systems: bid call (the auctioneer begins the auctions by notifying bidders of its start), bid collection (the auctioneer gathers all the bids), winner determination (the auctioneer selects and notifies the winning bidder). Furthermore, the operations are executed in the following order: first an auctioneer presents the item(s) to the auction participants (bidders), the bidders submit their bids, and then the auctioneer determines the winner. Auctions can be classified according to certain properties governing goods, bidding, and the eventual price the winner pays. Parsons *et al.* [62] states that auctions can be single-dimensional or multi-dimensional, one-sided or two-sided, open-cry or sealed-bid, first price or k-price and and single-unit or multi-unit. In single-dimensional auctions, auctioneers and bidders are only concerned with the price of the goods. In a multi-dimensional auctions, other features or characteristics of the goods come into play. In one-sided auctions, either the buyers (bidders) or sellers bid but not both, as in the case of a two-sided auction. In open-cry auctions, bid amounts are known by all bidders, and in sealed-bid, only the auctioneers know bid values. In a first price auction, the winner of the auction pays the winning bid; but in a k-price auction, the winner pays the kth bid, where bids are ranked in ascending or descending order. Lastly, assuming a unit of a good(s) has been defined, then in a single-unit auction, only one such unit is up for auction at a time; and in a multi-unit auction, buyers may bid on more than one unit. Multi-unit auctions are more commonly known as combinatorial auctions. In multi-robot systems, *items* can be tasks, targets or resources [61]. However, markets are not reserved solely for multi-robot systems; markets have also been employed in a variety of traffic control and management systems [63–65].

3.5 Market-Based Traffic Management

The use of economic principles in traffic management and control is nothing new, e.g., tolling systems have been in use for hundreds of year. Market-based traffic controllers (and traffic management systems) employ a variety of auctions, e.g., second price sealed bid [64], or combinatorial auctions [66] and Walrasian auction [65]. Auctions are a versatile framework for agent interaction and have woven their way into the traffic domain through a variety of approaches. Many market-based traffic systems treat roadways and/or intersections as a commodity where drivers participate in a market to gain access to said commodity. For example, Carlino *et al.* [64] implemented a traffic control system where vehicles bid on traffic signal phases. Schepperle and Böhm [67] employed a similar strategy except vehicles bid on periods of time when it is safe to traverse the intersection. While Carlino *et al.* [64] and Schepperle and Böhm [67] used the intersection as a commodity, Vasirani and Ossowski [66] presented a system for traffic assignment where an auction is used for route selection. One of the more interesting properties of auction-based traffic control systems is their ability to seamlessly incorporate individual valuations of time [63, 64, 67, 68].

Carlino *et al.* [64] described a traffic control system where auctions are run at intersections to determine use. Vehicles are embedded with a software agent (the *wallet agent*) which bids on behalf of the driver. A *system agent* also bids in a manner that facilitates traffic flow beneficial to the entire transportation system. However, the *wallet agent* is solely concerned with getting its occupants to their destination in the least expensive (and quickest) way. The *wallet agent* is assigned a budget to pay for trips. Carlino *et al.* [64] used a second-price sealed bid auction mechanism.

Carlino *et al.* [64] tested four different modes: *FIFO* (this is how a typical intersection works), *Equal* (every driver submits a bid of one), *Auction* (drivers bid an amount equal to their account balance divided by the number of intersections remaining on their trip), and *Fixed* (drivers always bid the same amount based on the value they've assigned for the trip). The authors evaluated their traffic control mechanisms in four simulated urban cities. FIFO performed the worst in three of the four cities. *Auction* (with and without the *system agent*) had the best performance.

There are two important distinctions between my work and [64]. First, Carlino *et al.* [64] assumes vehicles have specialised software that allows drivers to effortlessly participate in the auction; I do not need require any such software. Second, although auctions are used in my work, the auction provides a framework for coordination and is not monetised.

Schepperle and Böhm [68] described an auction-based traffic controller, *Initial Time-Slot Auction* (ITSA), where vehicles bid for a *time-slot*. A time-slot is a window in time when the vehicle may safely use the intersection. In ITSA, an *intersection agent* continually executes a second-price sealed-bid auction of the most current available time slot. A vehicle does not participate in an auction until all the vehicles in front of it have been allotted a time-slot. Schepperle and Böhm [68] implemented two versions of ITSA. In the first version (ITSA), auctions are suspended if any vehicle's wait time has reached a specified limit; this prevents accidental starvation of roadways. The second version allows subsidies (ITSA+Subsidies); vehicles that have not participated in an auction yet can influence the auction of the vehicles in front of them by subsidising the bidding vehicle.

Schepperle and Böhm [68] evaluated their approach using a single intersection (in simulation). Schepperle and Böhm [68] used *waiting time* to measure performance. The authors defined waiting time as the difference between actual travelling time and the minimum travel time. Schepperle and Böhm [68] also examined *average weighted waiting time* where the weighted waiting time is the product of the waiting time and the driver's valuation of a reduced waiting time. They compared their traffic controller to the reservation-based system in Dresner and Stone [51]. Both ITSA and ITSA+Subsidies were able to reduce average travel time while minimising average weighted waiting time compared to FIFO, although ITSA+Subsidies was better at reducing average weighted waiting time. Drivers that had the lowest valuations, that is those drivers that did not mind waiting, fared better under ITSA+Subsidies than ITSA.

In follow-on work, Schepperle and Böhm [67] created a *valuation-aware* traffic-control mechanism which allows concurrent use of the intersection through an auction mechanism. In a valuation-aware traffic controller, the intersection takes into account the driver's value of time; but many of these systems do not allow concurrent use of the intersection. Schepperle and Böhm [67] proposed two auction-based mechanisms: *Free Choice* and *Clocked.* In Free Choice, the auction winner gets to select the time slot it wants from an interval; while in Clocked, time slots are auctioned off. In Free Choice and Clocked, subsidies [68] are allowed because without subsidies vehicles with high valuations (vehicles that are pressed for time) may become trapped behind vehicles with low valuations of time. Schepperle and Böhm [67] also implemented a communication protocol, Time-Slot Request (TSR), which allows vehicles to request a time slot to traverse the intersection. TSR is similar to Dresner and Stone's reservation-based system [51] and is used to evaluate the performance of Free Choice and Clocked. Schepperle and

Böhm [67] included three levels of concurrency in their experiments: intersection exclusive (IE), lane exclusive (LE) and conflict area exclusive (CAE). In IE, only one vehicle can utilise the intersection at any moment in time; and in LE, two vehicles can use the intersection, as long as the two manoeuvres do not conflict with one another. Lastly, CAE, which is the highest level of concurrency, allows vehicles to perform manoeuvres as long as there is not more than one vehicle in any area of conflict. Schepperle and Böhm [67] concluded that Free Choice reduced the average weighted wait time by up to 38.1%. Clocked reduced the average weighted wait time for only lower degrees of concurrency and high traffic volume.

The major drawback with Schepperle and Böhm's work is that their auction-based traffic control systems require vehicle agents [67, 68]. Schepperle and Böhm assume that vehicles possess a device which provides a standard interface [68] with the transportation infrastructure. Furthermore, this device would require the installation of necessary agent-based software for the traffic control systems in [67, 68]. The great cost of including vehicle agents (and developing a standard $V2I$ interface) has been completely overlooked.

Iwanowski [69] proposed a multi-agent market-based traffic management system to balance road use. The traffic management system provides route guidance for drivers in order to reduce traffic congestion. The aim of [69] is to coordinate drivers via a central unit (the road network operator) and distribute the drivers across the network while taking into consideration individual route preferences.

Iwanowski decomposed the traffic management system into several agents: *vehicle agent*, *coordination agent*, *segment auctioneer agent* and *net communication agent*. The net communication agent provides information about traffic conditions to all the other agents; more specifically, it generates a set of routes for the vehicle agents. Vehicle agents, after receiving a set of routes to their destinations, bid on behalf of the driver for the most preferred route. Drivers are then expected to follow the route the vehicle agent has won, although, drivers are not forced to follow the route. Along with available routes, the net communication agent also includes additional information such as estimated route travel times and traffic conditions. This information is used by the vehicle agents to better asses the quality of a path. The coordination agent and the segment auctioneer agent facilitate the auction.

Iwanowski tested his approach on DARE (Distributed Agents Runtime Environment) using a variety of networks that had between 250 and 1200 road segments. Iwanowski employed six types of drivers that differed in route evaluation and bidding strategy. Additionally, it is assumed all drivers followed the route suggested by the vehicle agent. Iwanowski studied the effects that different levels of participation in the auction schema had on travel time and congestion. In the controlled experiment, no drivers used the auction-based route guidance system. Iwanowski measured travel times and congestion but, did not go into detail about how the latter was measured. The author found that there was a noticeable improvement in traffic flow when at least 30% of drivers utilised the guidance system. Also, the author found that a 50% participate rate worked nearly as well as having all drivers participate in the guidance system.

Again, as with many other auction-based traffic control and management systems, Iwanowski [69] relies on vehicle agents. Additionally, the role of the *net communication agent* suggests that devices, those used to collected traffic data, must also be connected to the transportation infrastructure or, at the very least, be connected to a communication hub of some sort. The author does not provide much detail as to how routes are selected for individual drivers. Lastly, Iwanowski [69] does not mention how traffic signal timing may influence the overall performance of the system, that is, what assumptions should be made about signalised intersections that are traversed in the system or whether this is another factor already used by the net communication agent for route selection.

Iwanowski's work in auction-based route guidance systems is further developed in [70]. Road signs (static or dynamic) and radio broadcast are still widely used by transportation departments to help drivers avoid congestion or traffic accidents. Although, navigation systems and mobile phone apps are becoming more popular choices for guidance, Iwanowski *et al.* [70] points out that these solutions (road signs and radio) are are overly simplistic and if all drivers were to respond in the same manner, it could just create traffic congestion in some other portion of the road network. Iwanowski *et al.* [70] present three auction-based guidance systems: auction-based trading, exchange-based trading and vehicle-to-vehicle transactions. The first two tackle dynamic route assignment and the third addresses *individual road clearance*. Individual road clearance is the process of clearing a lane in order to maintain speed or pass slow moving vehicles.

In auction-based trading, the vehicle agent (referred to as *vehicle/driver units* in [70]) plays a similar role to vehicle agents in [69], that is, vehicle agents bid for routes on behalf of the driver. The vehicle agent does not control the vehicle, thus, the vehicle agent is more or less a navigation system. The auction-based trading system also includes an auctioneer whose job it is to auction road segments to the drivers. Availability of roadways are based on the maximum capacity of each roadway. However, unlike in [69], in the auction-based trading system, drivers that have attained their preferred route pay drivers that are experiencing congestion.

In exchange-based trading, a *call market* is used to allot road segments. A call market is a type of exchange or double auction. In a call market, sellers set minimum prices for their goods while the buyers set the maximum price they are willing to pay for said goods. The price that results in the most goods being sold is set as the selling price of the goods. In [70], the goods represent access to road segments at specific time slots. Minimum prices are set by the auctioneer based on current traffic conditions. If a vehicle enters a road segment it does not have the right to, it is charged a fee.

In vehicle-to-vehicle transactions, passing vehicles pay the other vehicles that they have passed. This third guidance system only works if the road has at least two lanes and vehicles are able to transmit telemetric data. Iwanowski *et al.* [70] presented two variants of this idea: *bilateral trading* (the passing vehicle only pays when the other vehicle has to change lanes) and *fixed passing rate* (the passing vehicle always pays a fee to any vehicle it passes). Unfortunately, Iwanowski *et al.* [70] did not evaluate their proposed auction-based guidance system; thus, there is not indication of how well it would perform. Additionally, the authors state that the mechanism would be voluntary; but this is a major problem for traditional traffic coordination schemes. The author do not explain how to incentivise the system to increase participation. Without some form of incentive for drivers to utilise the mechanism, there is no guarantee it will improve traffic conditions. Although the guidance systems described in [70] includes a vehicle agent, the authors mention that the vehicle agents can either run on the vehicles or be centralised but accessible to the transportation infrastructure. However, the latter does not necessarily fit with all three mechanisms that were described, for example, the telemetric data that is required for vehicle-to-vehicle transactions. It is not clear how this data would be attained without embedded vehicle agents or at the very least $V2V$ and $V2I$ communications.

Isukapati and List [63] presented a multi-agent auction-based traffic control system which is akin to a toll system for road use; municipalities earn money from drivers using the road network. Similar to [64, 67], the traffic control system incorporates the driver's valuation of time (or VOT [63]). That is, drivers with high VOT are considered to be in a rush and can influence the behaviour of the traffic controller for their own benefit. In the traffic control system, at every intersection there are agents, *movement managers*, that represent the different possible traffic manoeuvres (e.g., left turns). Additionally, there is a driver agent which interacts with the transportation infrastructure on behalf of the driver. Movement managers receive fees and contributions from the drivers within their assigned traffic stream(s). The fee is a one-time payment to the movement manager as the vehicle approaches the intersection. The contributions however, are based on the drivers' VOT. Driver agents with a high VOT are more willing than drivers with low VOT to contribute to the movement managers. Driver agents also monitor their wait time and how well the movement manager is performing. The driver uses this information to determine when (and how much) to contribute. An auction is utilised to determine which movement manager will use the intersection. During auctions, movement managers submit their bids for the intersection; this is how revenue is generated. The winner of the auction will either receive a minimum amount of green time or if that movement manager already had access to the intersection, its green time will be extended.

Isukapati and List evaluated their approach under three different traffic scenarios. The traffic scenarios varied in the intensity of traffic heading north versus traffic heading east. The authors measured performance in a number of ways: payments made by the drivers, bids submitted by the movement managers, revenue generated by all intersections, and the delay experienced by drivers. In the first scenarios, there was an equal amount of traffic flowing north and east, while in the second and third scenarios, there was increasingly more traffic heading north. Isukapati and List compared their approach to gap-based actuated traffic signal controllers. Isukapati and List found that drivers with high VOT contributed more than drivers with low VOT. In terms of the bidding amount, as the disparity between north and east traffic increased, the heavy flow (heading north) generated smaller bids and the lighter flow (east) generated large bids. High VOT drivers contributed less in the third scenario than high VOT drivers in the first scenario. Drivers with high VOT in low flows ended up contributing more in the third scenario than in the first scenario. Isukapati and List found that the municipalities earned more in the first scenario than in the third scenario. Isukapati and List also found that their approach reduced the delay.

The authors' agent structure is similar to the one used in my work. More specifically, in this thesis, groups of vehicle manoeuvres are represented by agents. Furthermore, those agents participate in an auction to facilitate traffic management. However, there are a number of differences between the work of Isukapati and List and my approach. First, my agent bidding strategy is different from the one used in [63]. In [63], movement managers select one of three candidate bids. Second, driver agents must communicate with the movement managers which requires some form of $V2I$ technology. Additionally, the driver agents must also know their position in the vehicle queue. My auction-based traffic controller does not require driver agents or $V2I$ technology.

Vasirani and Ossowski's [65] approach to relieving traffic congestion has some similarities to Iwanowski *et al.* [69, 70]. Vasirani and Ossowski [65] utilised a competitive market to price roadways according to level of use. Pricing the roadways gives drivers an incentive to travel cheaper, less congested roadways. The end goal is to better distribute traffic across the entire road network. The multi-agent traffic assignment system in [65] is comprised of two types of agents: *vehicle agents* and *intersection manager agents*. The vehicle agents participate in the auction on behalf of the driver. The intersection manager is responsible for setting its price, more specifically, the price of traversing each of the intersection's incoming lanes. Vasirani and Ossowski used a Walrasian auction.

In a Walrasian auction, there are buyers and suppliers; here buyers are the driver agents and intersections supply access to roadways. The intersection manager agents set an initial selling price for each of their incoming lanes, after which the driver agents are notified of this initial price and determine their preferred route; driver agents always select the shortest route they can afford. This information, the routes the driver agents have selected, is shared with the intersection manager agents who then adjust the initial selling price. A new selling price is given by the following rule: *if there is an excess of goods, then the price is lowered; if there is an excess demand, then the price is raised* [65]. This process is repeated until demand matches supply, i.e., the number of drivers utilising an incoming lane matches the lane's capacity.

Vasirani and Ossowski [65] evaluated their approach in simulation on an urban-like road network. However, the road network included freeways as well as smaller roadways. Additionally, some of the intersections were controlled by traffic signals (intersections with three or more incoming lanes). The authors executed two experiments. In the first, the market-based system for roadway pricing was used to balance the distribution of traffic. In the second, only traffic signals were used without a market-based system to influence driver route selection. For evaluation purposes, the authors used several metrics: average travel time, average speed, the moving average of travel time and density. Vasirani and Ossowski [65] found that the competitive market reduced the minimum average travel time of drivers traversing the road network. Likewise, the competitive market reduced the moving average travel time and the road traffic density (i.e., vehicles were better distributed across the road network with the market than without it).

Unlike Iwanowski *et al.* [70], Vasirani and Ossowski [65] provide experimental results showing that their approach does in fact improve traffic performance. This paper however requires driver agents and other transportation features that are not yet available.

Vasirani and Ossowski [66] expanded on Dresner and Stone's [51] work by examining the performance changes to a reservation-based system where time slots were allocated using a *combinatorial auction (CA)*. The authors tackled both the traffic control problem (at a single intersection) and traffic assignment problem (in a road network) with their market-based approach. As drivers approached the intersection, reservations were awarded through the auction, instead of *first-come, first-served* (FCFS) which Dresner and Stone [51] employed in their work. In this way, drivers express their true valuation for a contested reservation. As stated earlier, Dresner and Stone's [51] traffic control systems relies on vehicle agents running on autonomous vehicles, and Vasirani and Ossowski's approach requires both as well. However, in Vasirani and Ossowski's [66] system, the vehicle agents are responsible for participating in the auction as well as controlling the vehicle.

In a network with a single intersection, the authors looked at the delay experienced by drivers based on the amount they were willing to "pay" to use the intersection. They found that initially having a willingness to pay does decrease delay, but eventually this levels off. However, CA was found to increase overall delay. As the intensity of traffic increased, CA experienced far more delays and rejected reservations than the FCFS approach.

For the traffic assignment system, Vasirani and Ossowski [66] devised a protocol where route selection was accomplished through *combinatorial traffic assignment* (CTA). The cost of passing through an intersection changed continually depending on traffic demand. Drivers could select their route, as they traversed the network, based on the path cost set by the intersections. In turn, these costs caused vehicles to select alternative or cheaper routes. An examination of the moving average travel time increase reveals the most with FCFS and the least with CTA. The authors concluded that CTA reduced average travel time but did not guarantee that drivers that paid more experienced less delay.

A key drawback with much of the literature on auction-based traffic control (and management) systems is their reliance on vehicle agents. Furthermore, the vehicle agents can be tasked with any number of responsibilities such as vehicle control (e.g., autonomous or semi-autonomous cars), facilitating payments, participating in auctions and/or engaging in communications with transportation infrastructure. The first issue is the development and distribution of vehicle agents. Car manufacturers will have to agree on international communication protocols, physical specifications and many other aspects of deploying vehicle agents to the millions of vehicles that are currently in use. However, this will not be easy considering the various roles vehicle agents currently play in auction-based traffic control and management systems. Second, there is the current state of the transportation infrastructure worldwide. The communication systems necessary for $V2I$ and $V2V$ communication currently do not exist; and a range of issues, such as security and privacy, remain unaddressed in the traffic management domain. This thesis proposes a multi-agent auction-based traffic control system which does not rely on vehicle agents. The aim of my work is to broaden our knowledge on the use of auctions in the traffic domain by imposing real-world constraints such as the absence of autonomous vehicle and $V2I$ communication capabilities.

3.6 Reinforcement Learning in Traffic Control

Mashayekhi and List [71] used a combination of Reinforcement-learning (RL) and auctions for traffic control. Mashayekhi and List's [71] agent framework is the same one used by Isukapati and List [63] which shares some similarities to my work (as mentioned above). Mashayekhi and List's [71] approach was used to investigate the effectiveness of my approach in Raphael *et al.* [16]. In this thesis, my auction-based traffic controller is compared with Bazzan *et al.*'s [32] RL-based traffic control system. This section covers a number of RL-based approaches to control assignment. RL is a machine learning technique where agents learn the value of their actions by exploring their environment [72]. Within the traffic domain, there is a plethora of RL-based traffic management systems that tackle traffic control and assignment.

Mashayekhi and List [71] designed a multi-agent auction-based traffic controller where the agents that participated in the auctions used RL to learn a bidding strategy. Mashayekhi and List [71] organised the different vehicle movements that take place at an intersection into groups, each represented by an agent (or *movement managers*). Movement managers participate in auctions to determine which pair of safe vehicle manoeuvres should be allowed to pass through the intersection. Auctions took place only when there was the possibility of a collision occurring between two or more vehicle manoeuvres. More specifically, an auction was executed whenever two or more roadways had vehicle queues. The movement managers used reinforcement learning to learn an optimal bidding strategy. Lastly, Mashayekhi and List [71] adjusted the phase order, more specifically, the order in which vehicle movements are made. That is, in [71], phases with low traffic volume can be skipped altogether, allowing the intersection to service a movement(s) with higher traffic demand.

Mashayekhi and List [71] evaluated their traffic controller with two traffic scenarios and used average travel time to measure performance. One scenario had low traffic volume and the other had a 25% increase in traffic volume. In the first scenario, their approach appeared to learn a policy that did better than the fixed-time traffic signals and actuated traffic signals. The learned policy failed in the second scenario with higher traffic volume. In the second scenario, the fixed-time traffic signals outperformed the auction-based traffic controller.

Although the traffic control system described by Mashayekhi and List [71] has an agent framework similar to my approach, there are many differences between my approach and theirs. First, my agents use a bidding strategy based on common traffic engineering practices, while Mashayekhi and List [71] used reinforcement learning to acquire a bidding strategy. Second, Mashayekhi and List's approach needs $V2I$ communication. As vehicles approach an intersection, they must report their presence to the movement managers via *tokens* [71]. My methods do not rely on any such technologies. Lastly, Mashayekhi and List's auction-based traffic controller also has dynamic phases. Meaning, there is no set sequence of phases because phase order is determined by the winner (movement manager) of the auction. However, in my approach, the phase order remains the same.

Bazzan *et al.* [32] presented another RL solution to traffic control (without the use of auctions). Bazzan *et al.* [32] proposed dividing the multi-agent system into smaller sub-networks to prevent an explosion in the joint action space which occurs when coordinating large groups of intersections. Bazzan *et al.* [32] utilised high-level agents or *supervisors* to observe small groups of local acting agents in search of an optimal joint policy. The *supervisors* collected information on the actions and rewards of their subordinates. The *supervisors* could then make suggestions, to the subordinates, about which actions to take. The subordinate agents, which represented intersections, always acted locally and could in some cases reject suggestions from their *supervisor*. The subordinate agents are also responsible for adjusting traffic signal timing. Bazzan *et al.* [32] discretised the intersection state-space to reduce the number of state-action pairs. Thus, the intersection can be in one of three possible states: the north/south bound traffic is greater than the east/west bound traffic, the north/south bound traffic is less than the east/west bound traffic, and finally, the north/south bound traffic is nearly equal to the east/west bound traffic. In [32], actions are entire cycle plans (each 60 seconds long). The subordinate agents have three possible actions, one to address the imbalance in each of the three states that have been described. Finally, the reward function is based on the estimated number of stopped vehicles, that is, the subordinate agents selected an action which reduced stops.

The authors evaluated their mechanism in ITSUMO [73], a microscopic traffic simulator. As a performance metric, the authors measured the total number of stopped vehicles in the network. The authors evaluated their system using two traffic scenarios and ran each scenario with and without *supervisor* agents. Similar to my experiments, the scenarios varied in traffic intensity levels. Under both traffic scenarios, the supervised version performed the best, i.e., had fewer stopped cars.

Another RL-based traffic controller was developed by Wiering [74]. In [74], each traffic signal is represented by an agent which independently learns which action (e.g., turn *green* or *red*) to take given its local traffic condition in order to reduce wait times. More specifically, traffic signal agents learn the wait time associated with each action using information from vehicles at the intersection the traffic signal agent serves. Furthermore, traffic signal agents must be able to communicate with vehicles and other traffic signal agents (other intersections). Vehicles share their positions in queues and their destinations with traffic signal agents. Using information from vehicles and other intersections, traffic signal agents are able to estimate vehicle wait times.

Wiering [74] evaluated three different levels of information sharing, that is, the information that traffic signal agents were allowed to send to one another. In TC-1, the traffic signal agents use only local information for computing wait times. In TC-2, traffic signal agents use information from other intersections to compute the wait time of the first vehicle in the queue only (for the other vehicles, local information is used). Lastly, in TC-3, traffic signal agents use information from other intersections to compute the wait time for all vehicles. Wiering [74] also includes an alternative version of each TC-1, TC-2 and TC-3 with a *co-learning* driving policy. RL-based traffic controllers with *co-learning* enabled allowed vehicles to utilise wait time information from traffic signal agents for path selection.

Wiering [74] compared the RL-based traffic controllers to three other traffic controllers: fixed-time traffic signals, random traffic signals (the traffic signals randomly select a state) and a traffic controller that always discharges the incoming link with the longest queue. The traffic controllers were evaluated on a grid-based network on a traffic simulator developed in-house. At each time step, one to eight vehicles were introduced into the road network and each vehicle had a randomly chosen origin and destination.

Wiering [74] found that TC-3 with *co-learning* outperformed the other traffic controllers when one vehicle is introduced in the road network per time step. Additionally, the random traffic controller performed the worst with one vehicle per time step. Wiering [74] also found that TC-3 with *co-learning* performed well when three vehicles were generated per time step. In the experiments where four or more vehicles were generated per time step, the author collected data on how many vehicles were rejected for entry into the network. That is, at each time when a vehicle is ready for entry into the network, it is possible that the roadway is filled and there is no room for a new vehicle. In situations where there is no room left and the network is saturated, the vehicle is rejected. Wiering [74] considered waiting time and the number of rejected vehicles when four or more vehicles were generated per time step. Wiering [74] found that when four vehicles were introduced per time step, TC-2 with *co-learning* had the lowest wait time and number of rejected vehicles. However, all three RL-based traffic controllers that did not use *co-learning* had longer wait times than fixed-time traffic signals when four vehicles were generated per time step.

Bakker *et al.* [47] described an RL-based traffic control system where traffic signal agents learn the appropriate phase to use for different traffic conditions. The traffic signal agents utilised local information about vehicles on the roadways to map traffic signal phases to average waiting time. Bakker *et al.* [47] implemented three traffic controllers: TC-SBC (State Bit for Congestion), TC-GAC (Gain Adapted by Congestion), and TC-SBC+GAC. In TC-SBC, the traffic controller considers congestion information from neighbouring junctions in the form of a single bit (i.e., 1 or 0) to minimise the increase in the state space. This single bit is part of the intersection's state representation, thus, allowing the traffic signal agent to learn a state transition function which considers neighbouring traffic conditions. In TC-GAC, the same congestion bit information from TC-SBC is incorporated in the traffic signal agents' action value function; as a result, there is no increase in the intersection's state space size. Finally, TC-SBC+GAC combines both ideas, TC-SBC and TC-GAC.

Bakker *et al.* [47] utilised average trip waiting time (ATWT) as the performance metric. The authors used another RL-based traffic controller, TC-1 [75], which does not include any form of intersection coordination, as a benchmark. Bakker *et al.* [47] conducted two experiments: one where vehicles were spawned at a fixed rate and another with a variable spawning rate (to simulate rush hour traffic). In the first experiment, TC-SBC+GAC had the lowest ATWT, while in the second experiment TC-GAC performed the best.

El-Tantawy *et al.* [76] presented another RL-based traffic control system which allows intersections to consider neighbouring traffic conditions in their decision making. The traffic control system, MARLIN-ATSC (Multi-agent Reinforcement Learning Integrated Network - Adaptive Traffic signal controller) has two modes: *independent* and *integrated*. In independent mode (MARL-I), the intersection agents act locally, i.e., the intersection agents do not take into consideration the actions of the other agents when deciding on an action. In integrated mode (MARLIN), the intersection agents coordinate their actions with neighbouring intersections. In the integrated mode, intersection agents learn a joint policy with their neighbours using modular Q-learning and Markov games. In MARLIN, the traffic control problem is divided into pairs of intersections agents. An intersection agent learns a set of Q-values for all possible state-action pairs between itself and its immediate neighbours. More specifically, each intersection agent plays a Markov game with each of its neighbours to learn the relationship between its phase and vehicle delay.

El-Tantawy *et al.* [76] used several metrics to evaluate performance, as well as vehicle delay. El-Tantawy *et al.* [76] collected data on average CO_2 emissions and average travel time. El-Tantawy *et al.* [76] used a microscopic traffic simulator to model the downtown area of the City of Toronto, ON, Canada and used signal timings provided by the City of Toronto. The area that was modelled used a combination of fixed-time signals, semi-actuated controls and SCOOT. El-Tantawy *et al.* [76] found that MARLIN-ATSC (in either mode) outperformed the benchmark. MARLIN showed the greatest improvement in delay and CO_2 emissions. The authors believe the intersection agents were able to *meter* (or gating) heavy traffic from upstream intersections to downstream intersections.

Another RL-based traffic controller which managed traffic by manipulating phase order was propose by Prabuchandran *et al.* [77]. Similar to [76], each intersection is represented by an agent which utilises Q-learning, however, Prabuchandran *et al.* [77] used a different approach for coordinating intersections. In Prabuchandran *et al.* [77], intersection agents utilise local information and congestion information from their neighbours to learn the appropriate phase order and coordinate their actions with their neighbours' actions. Prabuchandran *et al.* [77] investigated two exploration strategies: ϵ-greedy (Q-ϵ-greedy) or upper confidence bound (Q-UCB) [78]. Q-ϵ-greedy is the traditional method of exploration where a random action is selected with probability ϵ. In Q-UCB, an action is selected that maximises the difference between the Q-function and a term which represents an upper bound. This upper bound term is greater for actions that have been rarely played than actions that have been selected more often.

Prabuchandran *et al.* [77] compared their approach to fixed-time traffic signals (FST) where signal timings are optimised offline and saturation balancing (SAT) [79]. In addition, two versions of the RL algorithm with round-robin for phase selection were utilised, one with UCB (Q-UCB-RR) and the other with ϵ-greedy (Q-ϵ-greedy-RR). Prabuchandran *et al.* [77] evaluated their approach on the VISSIM traffic simulator on two road networks. Additionally, Prabuchandran *et al.* [77] used *average stopped delay*, *average delay* and *average number stops* as performance metrics.

Prabuchandran *et al.* [77] found that adjusting phase order reduced all three performance metrics on both road networks compared to FST, SAT, Q-UCB-RR and Q-ϵ-greedy-RR. Lastly, Prabuchandran *et al.* [77] also found that the *average stopped delay*, *average delay* and *average number stops* for Q-UCB and Q-ϵ-greedy were similar.

Khamis and Gomma [80] developed a multi-objective RL traffic control system aimed at reducing trip and waiting times. The RL traffic control system utilises a hybrid exploration approach, a combination of ϵ-greedy strategy and soft-max action selection. Khamis and Gomma's [80] approach utilises two types of agents: *vehicle agents* and *junction agents*. The vehicle agents represent vehicles and provide junction agents with information about current traffic conditions. The junction agent is responsible for setting appropriate signal timings given the information it receives from vehicle agents. Khamis and Gomma's [80] approach is similar to Wiering's [74] approach in that both traffic control systems rely on vehicle state data to estimate vehicle wait times.

Khamis and Gomma [80] combined all the reward functions (one for each objective) into a single Q-function, a summation of terms representing: *average trip waiting time* (from [74]), *average trip time*, *average junction waiting time*, *flow rate* (queueing), a measure of *green wave* formations, *accident avoidance* and finally *moderate speed* which rewards efficient fuel consumption. Khamis and Gomma [80] evaluated their approach on the Green Light District (GLD) [75] traffic simulator using a grid-based road network.

As benchmarks, Khamis and Gomma [80] used the TC-1 controller [74] and two other approaches that are built into GLD: Self-organizing Traffic Lights (SOTL) [81] and a Genetic Algorithm-based method (GA) [75]. Khamis and Gomma [80] found that their multi-objective RL traffic control system outperformed the benchmarks, i.e., had lower average trip time, average trip waiting time and trip stops than TC-1, SOTL and GA. Also, vehicles that used the multi-objective RL traffic control system also attained higher average speeds during their trip.

Vasirani and Ossowski [82] described an RL-based method for pricing reservations where vehicle agents were charged for reservations. Before beginning a trip, the vehicle agent calculates its intended route. As the trip progresses, the vehicle agent purchases reservations and re-calculates its route based on current reservation prices. Reservation prices are used to influence route selection, i.e., vehicle agents prefer less expensive routes. Furthermore, in [82] intersection managers coordinated their prices to maximise global profit. Vasirani and Ossowski [82] balanced local and global objectives using a *difference reward* [83]. The difference reward allowed intersection managers to align their local objective with the global objective.

The authors evaluated their approach on a traffic model of a sub-network of Madrid, Spain and utilised average travel time as their performance metric. Vasirani and Ossowski [82] found that average travel time was lower with the pricing scheme in place. Additionally, the authors discovered that the vehicle agents covered more ground when using the market. The authors attributed this to a greater distribution of traffic with the market as opposed to without the market where vehicle agents selected routes strictly according to travel time.

The next section covers the use of AI and other methods in traffic control and assignment.

3.7 Traffic Management and AI

Using reinforcement learning to learn a policy for setting traffic signals is just one of many approaches to adaptive traffic control and assignment. Many other techniques have been investigated for use in traffic control, for example, fuzzy logic [44], neural networks [45] and evolutionary algorithms [84] have all been applied to traffic control. Each technique has its advantages and disadvantages but the goal is always the same: to overcome the unique challenges that are faced in traffic management.

3.7.1 Rule-Based

Rule-based systems are a form of knowledge representation that is used in traffic management. Rule-based systems share some common properties: have a knowledge-base in the form of *if-then* rules, the reasoning of the system is exposed to the user, therefore, responses can be explained and they have some sort of inference engine for *reasoning* [85].

Before computers were prevalent, traffic signals were controlled by electromechanical devices [86], but the traffic signal timings were developed by traffic experts. Traffic experts possessed a wealth of knowledge about signal timings and their effects on traffic flow. Hence, the idea of capturing the thought process of traffic experts was very appealing. Early traffic management systems relied on traffic models and a wealth of knowledge gained from transportation experts. Traffic flow is highly unpredictable and and can be distributed over a large area. Thus, it is difficult from a machine learning perspective to find patterns or relationships between traffic flow and those devices in place to regulate it. Learning traffic models or simply predicting traffic behaviour without prior knowledge is challenging. The use of transportation experts, specifically their unique understanding of the dynamics of traffic flow, can lead to practical solutions that, although may not be optimal, are known to work in the real world. Rule-based approaches can best be described as falling on a spectrum: on one end, there are expert systems and the other are systems that follow simple *common sense* rules [87–90].

Dinanga and Pasin [91] developed an intersection control policy that utilised Vehicular Ad hoc NETworks (VANETs) and $V2I$. The traffic control mechanism in [91] assumes there are autonomous cars (with $V2I$ capabilities). In [91], autonomous vehicles are represented as agents and form VANETS on approach to the intersection. Once a VANET is formed, agents use basic traffic control rules to determine which sets of vehicles can safely pass through the intersection. Thus, Dinanga and Pasin [91] does away with traffic signals altogether. For example, one of the rules is to allow the incoming roadway with the largest queue to discharge first. Dinanga and Pasin [91] evaluated several polices (including fixed-time traffic signals) and measured the *fairness* of the policy, i.e., less imbalance in throughput of north/south (NS) versus east/west (EW) traffic. Dinanga and Pisa [91] tested their approach using three traffic conditions: equal intensity for NS and EW, greater intensity on NS and finally, greater intensity on EW. Although the rules were simple, the authors found that under certain traffic conditions, they were more *fair* than fixed-time traffic signals. However, [91] relies on autonomous vehicles and communication technology that is not yet readily available.

Dion and Hellinga [92] described another rule-based traffic control system, albeit without agents, that has more characteristics of an expert system than [91]. Signal Priority Procedure for Optimisation in Real-Time (or SPPORT) [92, 93] is designed to minimise the delay of public transportation vehicles while preventing any adverse effects to general traffic. SPPORT utilises heuristic rules governing traffic events to produce an optimal signal-timing plan. Traffic events are traffic conditions that are either known to disrupt traffic or are exploitable. For example, the presence of a platoon (a convoy of vehicles) is an exploitable event in that a large number of vehicles can pass through the intersection during a small period of time. Hence, SPPORT would have a rule stating *if a platoon is detected on this link, then switch to this plan* [91]. Unlike Dinanga and Pasin [91], SPPORT does not rely on cutting edge technology for traffic information. Instead, SPPORT uses vehicle detectors to form a model of traffic flow on incoming roadways. Dion and Hellinga [92] evaluated SPPORT under twelve different types of traffic conditions (e.g., varying traffic intensity or prioritising transit vehicles). In simulations, SPPORT reduced the average delay of public transportation vehicles by 13% to up to 49% compared to optimal fixed-time traffic signals. Nevertheless, Dion and Hellinga [92] did not explore the effects of using different traffic control parameters would have on the performance of SPPORT. Dion and Hellinga [92] adjusted all three traffic control parameters whenever a new plan was put in place.

Although intersections do affect one another, it is still possible to develop a traffic control system that ignores this relationship yet produces acceptable travel delays, e.g., [91, 92]. However, this point of view is not shared by Lämmer *et al.* [94]. Lämmer *et al.* [94] contend that independent intersections, i.e., those that only act on local conditions, are inadequate. Instead, Lämmer *et al.* [94] proposed using *supervisors*, programs that monitor multiple intersections, to capture high-level traffic conditions. The supervisors follow a set of general rules to influence local signal-timing plans. Localised intersections with supervisors caused less delay for all forms of transport (e.g., buses and cars) than the state-of-the-art adaptive traffic control system it was compared to [94]. In [94], the supervisors set bounds or constraints on the possible signal timing adjustments performed by the local optimisers. Again, Lämmer *et al.* [94] did not investigate different traffic control parameters for use by local optimisers except for *green* time and phase order. Lämmer *et al.* [94] simulated a section of a road network in Germany during rush hour.

Martí *et al.* [95] proposed an expert system for traffic assignment under adverse weather conditions. The system was designed to provide traffic and weather data to drivers to improve route selection. The traffic management system was hierarchal and composed of a number of different types of agents that share responsibility for gathering and distributing traffic information. The proposed system had two modes of operation: coordinate and local. In coordinate mode, all the agents worked together to provide accurate road information while in local mode each division of agents can function independently. Martí *et al.* [95] did not provided an evaluation of their proposed system.

3.7.2 Fuzzy Systems

Fuzzy logic is a multi-valued logic system where a proposition may have many values. Similar to expert systems, fuzzy systems have (inference) rules. In fuzzy systems, solutions are encoded as a set of inference rules over fuzzy variables. Furthermore, the use of human-friendly linguistic variables and rules allows solutions to be developed in a human-friendly manner.

Abdelhameed *et al.* [48] described a multi-agent Fuzzy-Genetic (GA) hybrid for traffic control or Intersection Control System (ICS). In [48], GA is used to calibrate member functions. Intersection agents use fuzzy logic to determine instructions to send to autonomous cars —under the control of a vehicle agent. The fuzzy logic controller processes vehicle trajectories and then sends *driver* commands to autonomous cars for safe and speedy travel through the intersection. The hybrid system was able to reduce average delay time by 90% compared to fixed-time traffic signals. However, ICS assumes vehicles are autonomous and does not use traffic signals to carry out its control policy. Abdelhameed *et al.* [48] tested their mechanism on a single intersection. Although Abdelhameed *et al.* [48] does evaluate their approach using different driver behaviours (e.g., driver velocity), the authors do not try different kinds of traffic conditions.

Alam and Pandey [23] kept the traffic signals in their fuzzy logic controller. The fuzzy system in [23] utilised vehicle queues (from vehicle detectors) to determine whether to extend or terminate the current signal phase. The fuzzy traffic controller was able to cut the average vehicle delay nearly in half compared to a fixed-time controller [23]. However, other than signal phase transitions, Alam and Pandey [23] did not probe other types of signal timing adjustment (i.e., other traffic control parameters). Additionally, the traffic controller was tested using a single traffic condition.

Chiu [44] highlights one of the key advantages of rule-based systems: the human-friendly format of rules allows for easy porting of knowledge from one system to another. Chiu [44] designed a fuzzy system to adjust the *cycle*, *split* and *offset*. The fuzzy rules were based on the same principles as those used by SCATS. Chiu [44] evaluated their traffic controller on a small road network (in simulations). The author found that when all intersections used the fuzzy traffic controller there was significantly less waiting time and vehicle stops. The fuzzy system in [44] fine-tuned all three of the traffic control parameters that are studied in this thesis. However, Chiu [44] did not analyse the traffic controller with different combinations of traffic control parameters. Lastly, Chiu [44] used a single traffic scenario for testing.

Milanés *et al.* [96] developed a traffic control system similar to the traffic control system described by Dinanga and Pasin [91]. The traffic control system in [96] requires autonomous vehicles with $V2V$ and $V2I$ capabilities. In [96], vehicles communicate their position to one another (over a peer-to-peer WiFi network). Once the fuzzy system has all the positional information it needs, it will then broadcast commands to the vehicles within its range. The commands control the throttle and brake, i.e., vehicles will either increase/decrease or maintain their current velocity. Moreover, the commands ensure vehicles traverse the intersection collision free. Milanés *et al.* [96] used technology that is not yet available, e.g., $V2V$ and $V2I$, as well as autonomous cars.

Shahraki *et al.* [97] designed a fuzzy signal control system parameterised over phase selection. In other words, the fuzzy controller determined whether to extend the current phase or move to another. The fuzzy controller utilised vehicle queue lengths, vehicle counts and wait time for input. Although Shahraki *et al.* [97] did use traffic signals, the authors focused on a single method of signal timing adjustments (phase selection). Shahraki *et al.* [97] evaluated their traffic controller using a single type of traffic condition and found better throughput with the fuzzy traffic controller.

Similar to Shahraki *et al.* [97], Tan *et al.* [98] presented a fuzzy traffic controller with phase selection as its sole method of traffic control. The fuzzy traffic controller used vehicle counts and queue lengths as input. Tan *et al.* [98] used a single fuzzy controller that determined how long the current phase should be extended, if at all (Shahraki *et al.* [97] used two fuzzy controllers, one to determine the length of the extension and the other to measure the roadway's priority level). Additionally, like many other papers, Tan *et al.* [98] did not examine any other means of adjusting signal timing and relied on a single type of traffic condition for testing.

3.7.3 Genetic Algorithms

Genetic Algorithms (GA) based traffic control systems evolve optimal signal timing. GAs are algorithms that find optimal solutions through an evolutionary-like process. An initial population of solutions is created and evolved (via mating and mutations) into better solutions. GA-based approaches to traffic control tend to optimise several intersections together; each individual in a population is a solution to the problem. The value in this is that signal timings are coordinated, i.e., intersections will appear to work together. In general, GA traffic control systems work in the following manner. First, an initial population (or solution set) is created. Second, the population is evaluated (based on how well the solution improves traffic conditions). Finally, members of the population mate (and have their genes mutated) to create a new generation and the process begins anew.

Balaji *et al.* [99] used a GA to develop a centralised multi-agent traffic control system. A central controller used a GA to determine the best green times for intersections to use. The agents, each one representing an intersection, were responsible for collecting data for the GA system. The central controller gathered information from intersection agents at fixed intervals. The fitness function equated *wait time* in terms of arrival flow rate, saturated flow rate (rate at which vehicles left the intersection) and the current amount of *green* time. The GA system was able to reduce delay across a simulated network with six junctions. Although Balaji *et al.* [99] used traffic signals, the authors only studied the efficacy of *green* time adjustment and did not examine other traffic control parameters (e.g., *cycle length*) and other traffic conditions.

Ceylan and Bell [100] proposed a unique GA system, GATRANSPFE (a combination of GA, TRANSYT [26, 101] and the PFE) which tackles traffic control and assignment. GATRANSPFE uses two popular traffic models: TRANSYT, the precursor of SCOOT and PFE (Path Flow Estimator). The GA is used to evolve traffic signal timings. TRANSYT is used to assess signal timings (i.e., it is used to determine the genetic fitness of signal timings). However, TRANSYT requires traffic flow data which is the purpose of PFE —estimate traffic flows given current traffic conditions. In terms of traffic control parameters, GATRANSPFE optimises all three traffic control parameters, *green* time, *cycle length* and *offset*. Ceylan and Bell [100] did not examination different combinations of these traffic control parameters and only used a single traffic pattern.

Foy *et al.* [84] described a traffic signal controller that uses GA to evolve *green* times. Foy *et al.* [84] used estimated delay as a fitness function. Unlike Cyelan and

Bell [100] which used traffic models to evaluate signal timings, Foy *et al.* [84] used a traffic simulator. In other words, to determine the fitness of a population, Foy *et al.* [84] executed a traffic simulation using the signal timings (from the population). The traffic simulations allowed Foy *et al.* [84] to estimate the fitness or amount of delay caused by each possible signal timing. Foy *et al.* [84] relied on vehicle detectors at each intersection (currently available technology). However, like many other traffic control papers, it focuses on a single traffic control parameter and traffic pattern for testing. Similar to Foy *et al.* [84], Stevanovic *et al.* [24] utilised a traffic simulator as an evaluation function in its GA-based traffic controller. However, Stevanovic *et al.* [24] optimised *cycle length*, *green* time, *offset* and phase/stage sequence. The traffic simulator used in [24] has a number of measures of performance which the authors used as fitness functions (e.g., vehicle delay or average number of stopped vehicle). Stevanovic *et al.* [24] employed adjustments to all the parameters and did not test the efficacy of adjusting different combinations of control parameters nor did the authors investigate multiple traffic patterns.

Hercog [102] proposed using a Learning Classifier System (LCS) for traffic assignment. LCS merges reinforcement-learning and evolutionary algorithms to form a rule-based machine learning technique. In [102], a GA was used to continually evolve the rule-base. Hercog [102] developed the system to use historical weather and congestion data to provide travel suggestions to drivers.

3.7.4 Artificial Neural Networks

Artificial neural networks are a class of machine learning techniques inspired by biological brains. Araghi *et al.* [46] proposed a multi-agent Fuzzy/Neural network hybrid system for traffic control. The authors used a multi-layer neural network to fine tune the fuzzy parameters. Normally, membership functions are predefined using prior knowledge. Araghi *et al.* [46] contends that neural networks can be used to learn membership functions in lieu of pre-set membership functions where there is little or no prior knowledge. In [46], each intersection is represented as an agent; agents are selfish and do not consider the effect of their actions on their neighbours. Intersections use queue lengths and incoming traffic volume to define traffic state. The Fuzzy/Neural network hybrid traffic control determines the best *split* to use for the next stage/phase. Araghi *et al.* [46] did not investigate other traffic control parameters and their effects on traffic performance. Also, only a single traffic pattern was used during testing.

Choy *et al.* [103] also blend fuzzy systems with elements of neural networks. Their hierarchical multi-agent traffic control system pairs an intersection agent with each intersection in the road network and employs *zone* agents that control small groups of intersections agents. It is the *zone* agent that receives traffic volume data (from vehicle detectors located at each intersections under its control) and processes this information. The fuzzy-neural system returns signal timing policies that subordinate intersections must adopt. The fuzzy-neural system is capable of adjusting *split*, *cycle length*, and *offset*, but Choy *et al.* [103] did not study these traffic control parameters individually or in pairs. Choy *et al.* [103] used two traffic conditions (the first with a single spike in intensity and the other with two) for testing. Under both patterns, the multi-agent Fuzzy/Neural hybrid system reduced delay.

Spall and Chin [45] used a neural network to learn ideal signal timings for any given traffic condition. The traffic control system produced system-wide adjustments, i.e., the controller attempted to learn a control strategy for the entire road network which is far more difficult than producing a solution for a single intersection. The neural network was fed information on current traffic conditions and determined signal timings to alleviate issues such as congestion or adverse weather. The traffic controller also used vehicle detectors to measure traffic flow. However, there was not a comprehensive study of the specific traffic control parameters. The neural-based approach in [45] did show improvements in intersection throughput under a single type of traffic pattern.

3.7.5 Intelligent Transportation Systems

While all the techniques mentioned above cover computer science fields, there are more general uses of computing in transportation. Intelligent Transportation Systems (ITS) uses information technology (IT) to improve transportation networks, e.g., increase the efficiency of road networks or driver safety. However, ITS is a very broad term that encompasses the transmission, storage and processing of data. Examples of ITS include the use of databases, mobile computing devices, GPS/GSM, navigation systems, and smart transportation signs. In general, ITS seeks to take advantage of any technology, e.g., $V2I$ and $V2V$, to improve transportation systems. For example, ICA (Intersection Collision Avoidance) is a system that provides warnings (e.g., auditory) in cases where a vehicle is approaching an intersection at an unsafe speed and/or trajectory [53]. Although ITS does cover more common use of computing tools in transportation, there is some overlap with areas that have been traditionally found in computer science fields. The overlap between computer science and ITS has led to the development of other systems including Collaborative Driving Systems (CDS), Advanced Vehicle Control and Safety Systems (AVCSS) and Automated Vehicle Control Systems (AVCS), Adaptive Cruise Control (ACC) and Cooperative Adaptive Cruise Control Control (CACC) [41]. These systems depend on human drivers and more or less serve to augment human senses. However, they utilised some of the same technology found in autonomous cars. For example, some use computer vision, lasers and (inter-vehicle or infrastructure) communications.

3.8 Conclusion

Traffic management has been tackled using a variety of methods. Researchers have investigated the use of markets, fuzzy logic and machine learning techniques such as reinforcement learning and artificial neural network to improve traffic flow. However, this thesis is most concerned with market-based approaches to traffic control. Market-based traffic management systems have been shown to improve traffic performance. However, current methods of employing markets within the traffic domain require vehicle agents. Most often the vehicle agents are specialised software which perform vital market operations (e.g., bidding). Using markets in this manner, more specifically, where software agents represent the human driver and participate in the auction on behalf of the driver, creates a major hurdle for the deployment of such systems in the real-world. In other words, current approaches to market-based traffic management are not practical. Markets mechanism are flexible schemes, but the literature on market-based traffic control systems does not reflect that flexibility.

The popularity of traffic signals means that there is an abundance of traffic control systems which utilise traffic signals for intersection control. However, many papers fail to explore the relationship between traffic control parameters and traffic performance within their approach. Some traffic control systems adjust three control parameters while others only adjust one, e.g., *split* with no explanation as to how other control parameters interact with the chosen decision-making component. The traffic control parameters define the solution space and play an important role in the effectiveness of the overarching systems regardless of the method used for decision-making. The work in this thesis is an attempt to begin to fill the gap in the application of markets in traffic control, as well as to investigate the relationship between traffic performance and traffic control parameters.

Chapter 4

Market-Based Traffic Control System

4.1 Introduction

Adaptive traffic control systems require reliable data in real-time that provides an accurate representation of traffic conditions. The data can be vehicle queue length, average travel times or some other measurement of performance which informs the decision-making processes of the traffic control system. The traffic control literature is filled with market-based approaches that overlook the technological advances that are needed to acquire such data. The reality, however, is that there are many limitations on the manner in which this traffic data is produced despite advances in communications and geolocation software and devices. At the present, the technology does exist to provide accurate traffic flow data, e.g., travel times. For example, GPS data from mobile phones can provide more accurate information on traffic flow. The transportation industry has recognised the possible benefits of fine grain traffic data and this has led to efforts to capture location data from mobile networks (e.g., Global System for Mobile communication) [104], however, these efforts have not fully matured and are not in wide spread use. Thus, there remains a gap in market-based approaches that are designed specifically for use within current transportation infrastructure.

The market-based traffic control system described in this chapter is more practical than other approaches to market-based traffic control systems. The proposed approach has the following advantages:

- Vehicle agents. My approach does not utilise vehicle agents or any embedded software systems found in other market-based traffic control systems that are responsible for such task as communications ($V2I$ and $V2V$) or navigation and control (e.g., autonomous cars);
- Traffic data. My approach relies on existing transportation devices for collecting data on traffic conditions. More specifically, it uses in-ground induction-loop vehicle detectors to measure traffic conditions;

- Traffic signals. My approach also utilises traffic signals for managing traffic flow. Traffic signals are a popular method of traffic control. Thus, the proposed approach utilises the manipulation of three traffic control parameters (*split*, *cycle* and *offset*) that have been used successfully to improve traffic performance in the past.

My approach to market-based traffic control is inspired by my work with multi-robot systems [105, 106]. More specifically, the use of auctions to enable coordination in multi-robot problems, such as task allocation and robot routing [107–110]. In market-based multi-robot systems, auctions are used to manage resource allocation, or to assign task (or targets) to a team of robots. In multi-robot systems, auctions provide a method of coordination that has low communication and computational cost. The bid is a compact message between auctioneers and bidders that encapsulates all necessary information about a bidder's valuation of an item (or particular allocation). Similarly, the bidding process happens in parallel, reducing the systems computational cost.

This chapter describes my four market-based traffic control systems each addresses a research question proposed in Section 1.4.

First, SAT (Saturation) and SATQ (Saturation with Queuing) address :

Research Question 1 *How can a market-based traffic controller function without on-board vehicle software (e.g., vehicle agent) or transportation infrastructure upgrades (e.g., communication devices)?*

Second, GRACE (**G**ene**R**al-purpose **A**uction-based Traffic **C**ontroll**E**r), addresses two questions:

Research Question 2 *How can the use of split, cycle and offset adjustments be used to improve traffic performance?*

Research Question 3 *How does adjusting split differ from adjusting cycle, offset or a combination of the three traffic control parameters affect the performance of market-based traffic control system?*

Lastly, DC2 employs dynamic coalitions and addresses:

Research Question 4 *How can intersections in the proposed market-based traffic control system expand their working boundary through the use of dynamic coalitions?*

4.2 Agent Framework and Auction

In my market-based traffic controller, the intersection is composed of two types of agents: *intersection agents* and *traffic signal agents*. At an intersection, there is a single intersection agent and multiple traffic signal agents (see Figure 4.1). The intersection agent

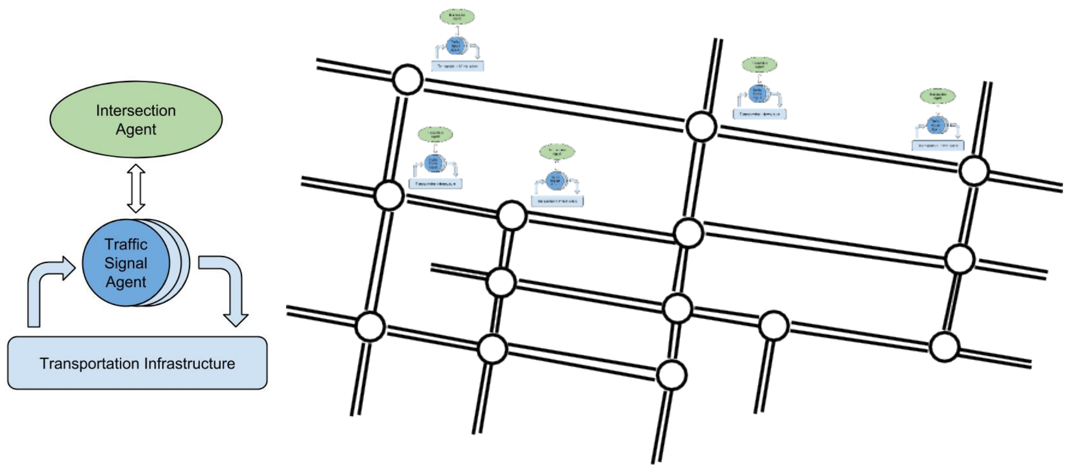


FIGURE 4.1: The agent framework for the market-based traffic controller. The intersection agents (also auctioneers) are responsible for making adjustments to traffic signal timings and executing auctions. Traffic signal agents, on the other hand, operate on behalf of a small set of legal vehicle movements that may occur at the intersection. The traffic signal agents compete against each other for control over traffic signal timing adjustments.

is responsible for making adjustments to traffic signal timings and ensuring that those changes do not violate any basic traffic regulations (e.g., minimum *green times*). Traffic signal agents, on the other hand, operate on behalf of a small set of legal vehicle movements that may occur at the intersection. That is, each traffic signal agent is assigned a number of movements to manage. The traffic signal agents compete against each other for control over traffic signal timing adjustments. An intersection agent and its associated traffic signal agents work together at the intersection level to adapt signal timings in real time. The adjustments are made to improve the efficiency of the intersection and maintain its safety.

The traffic signal agents are equivalent to traffic phases [22] in that they too represent a set of vehicle movements. Thus, for every phase in the phase plan, there is a traffic signal agent that functions on its behalf to tweak the time allotted to that phase. Together, all the phases form the signal timing for a traffic signal, while the traffic signal agents function as an intelligent counterpart to the phase. These two constructs, phase plan and traffic signal agents, address the needs of all legal vehicle movements as traffic demands change. The design guidelines set by traffic engineers for phase plans (e.g., in the U.S., they use MUTCD [18]) therefore provide a blueprint for determining which movements will be assigned to which traffic signal agent. Figure 4.2 illustrates the relationship between the traffic signal agents and the traffic phases. As there are two phases, there are also two traffic signal agents.

There is a natural conflict that arises between traffic signal agents assigned to an intersection. Each traffic signal agent is designated to a single phase in the traffic signal timing. They compete for a slice of the limited amount of available green time in a *cycle* (see Figure 4.2). Assuming the cycle length remains the same, giving more green time

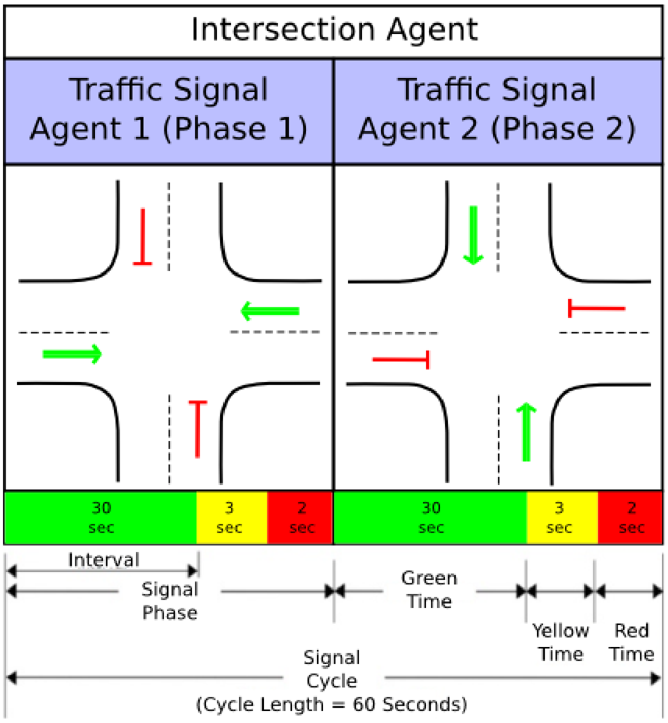


FIGURE 4.2: Traffic signal agents and their relationship to phase plans. For each phase, their is a corresponding traffic signal agent.

to one traffic signal agent means taking it away from another traffic signal agent. Thus, a multi-agent interaction protocol [13] is needed to determine an appropriate, adaptive allocation of green time to two competing entities.

As traffic flows through the intersection, auctions take place at fixed intervals which is called the *auction period*. The intersection agents serve as auctioneers and facilitate the auctions, that is, collect bids and determine the winner of the auction. The traffic signal agents participate in the auction and *bid* against each other to dictate how the traffic signal timing will be adjusted. The winner is the traffic signal agent with the highest bid. Note that the auction period does not have to match the cycle length. An auction may occur in the middle of a cycle or after a series of cycles have passed. Signal timings are only updated after the current traffic signal phase has completed. As a safeguard against starvation, traffic signal agents are prevented from having less than 10 seconds of green time. Using the taxonomy described by Parsons *et al.* [62], the auction used in my work can best be categorised as *single dimension*, *one-sided*, *sealed-bid*, *first-price* and *single-item*. Thus, the manner in which the auction is utilised in my market-based traffic control system resembles the *sequential single item* (a sequence of single-item auctions) auction in market-based multi-robot coordination [12, 111].

4.2.1 Vehicle Detectors

The most common type of vehicle detectors are inductive-loop detectors. Inductive-loop detectors are coils of wire laid into the ground. A small current is passed through the coils (or loop) to create an electromagnetic field. A vehicle (or any other large metallic object) that passes through the field will create a magnetic disturbance which signals a vehicle is present. Vehicle detectors may use any number of active and passive means to identify vehicle(s) within the detection zone, including video images, microwaves, lasers, radar, and infrared acoustics [112]. Vehicle detectors can be placed upstream from the intersection or downstream (near the intersection stop line). The traffic signal agents in my approach utilise vehicle detectors to assess road conditions and generate bids. Vehicle detectors provide estimations of traffic volume measured in vehicles per hour (*vph*) and vehicle counts. Traffic volume, measured by counting the number of vehicles N (reported by vehicle detectors) that pass a point on a road segment during time interval Δt [113], is computed as $v = N/\Delta t$.

If placed upstream from the intersection, vehicle detectors can also estimate the number of stops that will occur given the current signal timing and historical vehicle counts (the historical vehicle counts are always from the previous five minutes from the request for an estimate). The time-space diagram, shown in Figure 4.3, illustrates how the number of stops are estimated (the same method is used in SCOOT). The number of vehicles detected (upstream) from point T_{gc} to point T_{rc} in time will reach the intersection during a *red* interval; these are the vehicles that will have to stop. T_{gc} is the last time a vehicle can cross the detector during the green interval and make the light. And T_{rc} is the last time a vehicle will cross the detector and get stuck at the red light.

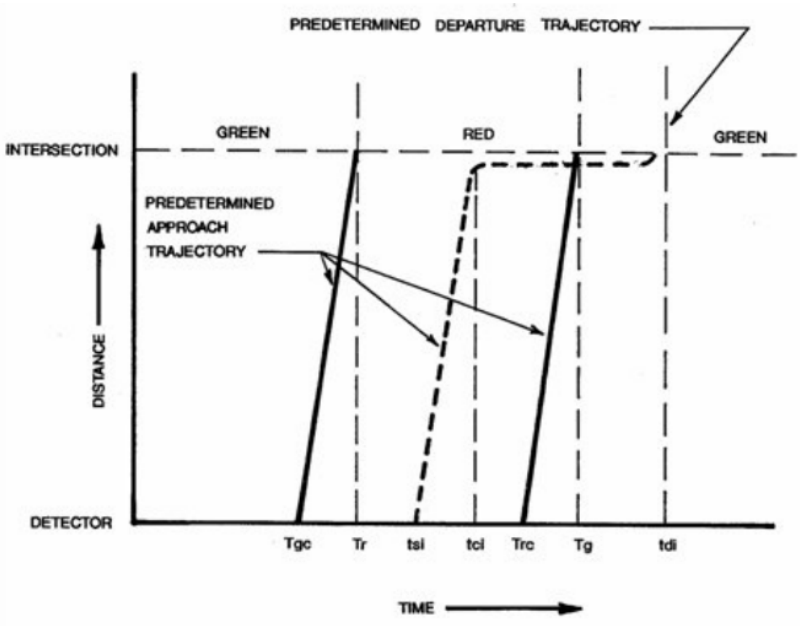


FIGURE 4.3: Time-space diagram for estimating stops. Vehicles that leave the upstream intersection (labelled detector) between time T_{gc} and T_{rc} will reach the downstream intersection when the phase is showing red [1].

4.3 SAT & SATQ

Traffic congestion occurs when the volume of traffic (i.e., number of vehicles passing a point per hour) exceeds the roadway's capacity and large vehicle queues begin to form. Left unchecked, vehicle queues will eventually spread to other intersections and what may have started at a single intersection spreads to other parts of the road network. Thus, it is vital to provide balanced access to the intersection, i.e., ensure that cross traffic demands are met. Saturation (SAT) and Saturation with Queuing (SATQ) [15, 17] are designed to prevent congestion from occurring. In SAT/Q[1], roadway *saturation*, a measure of traffic demand, is used to adjust the *split*. The rational for using saturation is that congestion (or over saturation) can be addressed, at the intersection level, by allocating more green time to the link(s) experiencing the greatest traffic demand.

SAT/Q is described in Algorithm 1. Traffic signal agents always begin with an initial traffic signal timing (lines 5 to 8) which is described in Section 5.7. In SAT/Q, auctions are executed periodically (line 12), every 5 minutes, it is only then that traffic signal timings are updated. Five minutes was chosen after a series of experiments similar to those described in [15]. Thus, every 5 minutes the auctioneer (or intersection agent) conducts an auction where the traffic signal agents bid against each other. The auction mechanism has two parts. First, the traffic signal agents submit their bids (line 15) and then a winner is selected (line 18). The traffic signal agent with the highest bid is always chosen as the winner (in case of a tie, the winner is chosen randomly). The bidding strategy used by traffic signal agents in SAT/Q are described in Sections 4.3.1 and 4.3.2. The winner of the auction gains 5 additional seconds of green time (line 19), while the loser's green time decreases by the same amount.

In SAT/Q vehicle detectors are located 20 meters from the intersection stop line (the hash-patterned and black rectangles in Figure 4.4). The vehicle detectors provide data on traffic volume which is used by the traffic signal agents to generate bids. More specifically, the traffic signal agents utilise the saturation of the road segments they serve. Saturation is defined as the ratio of traffic volume on a road segment to its capacity (this is considered a measurement of the level of use of a phase [15]). In general, a stream of traffic that is functioning closer to its capacity is more susceptible to traffic jams and delays [113]. Given a phase p which services K links (i.e., during phase p traffic is allowed to perform manoeuvres on K incoming links), let d_p be the measure of its saturation:

$$d_p = \sum_{k=1}^{K} \frac{v_k}{c_k}$$

where v_k and c_k are the traffic volume (measured by vehicle detectors) and *capacity* (the maximum possible traffic volume) on the link, respectively. Again, saturation is a measure of traffic demand on a link. The benefit of using saturation is that high levels

[1]SAT/Q is short hand for 'SAT and SATQ'.

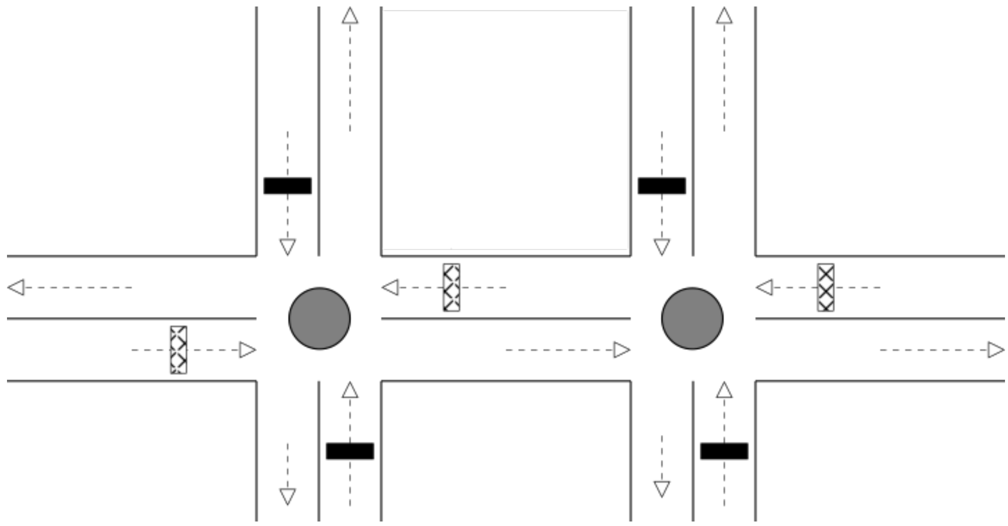


FIGURE 4.4: Traffic Signalling Scheme. The hash-patterned rectangles represent the pre-existing *induction-loop* sensors for the west/east traffic signal agents; black rectangles for the north/south traffic signal agents. Grey circles indicate intersection agents (though they have no physical embodiment in the simulated system).

of saturation can be an indicator that traffic congestion is imminent. Thus, saturation gives traffic signal agents a means of measuring the health of the link(s) it serves.

Two traffic signal agents were implemented which have different bidding rules: SAT (Section 4.3.1) and SATQ (Section 4.3.2).

4.3.1 Saturation (SAT).

In the SAT method, the traffic signal agents compute d_p for their road segment to use as their bidding rule. In the experiments conducted here, the traffic signal agents are only concerned with the single road segment preceding the junction they manage. For example, the west/east signal agent collects volume data one block west and one block east of its location. Equation 4.1 defines the SAT bidding rule:

$$bid = d_p \tag{4.1}$$

4.3.2 Saturation with Queuing (SATQ).

The SATQ method extends the SAT method, by augmenting its bidding rule with a measurement of the "fullness" of incoming links. The "fullness", u_p, of the incoming links serviced by phase p is the percentage of the roadway(s) that is occupied by vehicles (u_p is a value within the interval $[0, 1]$) where zero means the links are empty and one means the links are completely filled with vehicles. This provides a better picture of road conditions (e.g., whether there is a queue of vehicles leading up to the road sensor) than the saturation value alone. A traffic camera could be used to obtain this data.

Equation 4.2 defines the SATQ bidding rule:

$$bid = d_p + u_p \tag{4.2}$$

Algorithm 1
SAT/Q Traffic Control Algorithm.

```
t ← 0
H ← 15000                                   ▷ Maximum allowable time step.

J ← InitialiseIntersectionAgents()
for j ∈ Junctions do
    A_j ← InitialiseTrafficSignalAgents()
    T_j ← InitialiseTrafficSignalTiming()
end for

while true do
    for j ∈ J do
        if auctionTriggered() then          ▷ Determined by auction period.
            B ← ∅                           ▷ Collection of bids
            for k ∈ A_j do
                B ← submitBid(bid_k)        ▷ SAT uses Eq. 4.1 and SATQ uses Eq. 4.2
            end for

            k = determineWinner(B)          ▷ Winning traffic signal agent
            updateSplit(T_j, k)             ▷ Agent k receives 5 sec. of green time from loser.
            setTrafficLights(T_j)
        else
            setTrafficLights(T_j)
        end if
    end for

    t ← t + 1
    if t > H then
        terminateSimulation()
    end if
end while
```

4.4 GRACE

SAT/Q are limited in their ability to react to changing traffic conditions because only the *split* is adjusted. In SAT/Q the *intersection agent* auctions off 5 second segments of *green time* to the *traffic signal agents*. The winner of the auction gains 5 seconds of *green time* and the agent that loses the auction has her *green time* reduced by 5 seconds. More importantly, SAT/Q ignores other traffic control parameters that influence traffic performance. The most important components of traffic signal timing are *split*, *cycle*, and *offset* as demonstrated in commercial traffic control systems, such as SCOOT [26], RHODES [27] and OPAC [20, 28]. Furthermore, including *offset* adjustments would create new possibilities for intersection coordination where entire platoons are shuttled through multiple intersections with fewer stops. Thus, a different commodity is needed to allow *traffic signal agents* to manipulate all three traffic control parameters. This section presents GRACE [16], the market-based traffic control system which allows traffic signal agents to change all three traffic control parameters. In order to tune these parameters for real-time traffic control, GRACE addresses a number of questions: *Which parameters should be adjusted? When should the parameters be adjusted? What data is used to inform an adjustment?* and *How should the parameters be adjusted?*

The main difference between SAT/Q and GRACE is that in GRACE, the *authority* to make adjustments to traffic signal timing is the commodity; traffic signal agents no longer auction off just 5 seconds of *green time*. In GRACE, the winner of the auction is allowed to make changes to the intersection's traffic signal timing as she sees fit. This allows traffic signal agents to modify three important traffic control parameters that affect traffic performance. GRACE is described in Algorithm 2. As with SAT/Q, traffic signal agents begin with a default traffic signal timing (lines 5 to 9) —the same as those used in SAT/Q. Traffic signal agents in GRACE, generate a set, S, of all possible traffic signal timing adjustments (line 13). The set S is a collection of adjustment vectors where an adjustment vector $\mathbf{s}$ is defined as:

$$\mathbf{s} = \langle \delta g, \delta c, \delta t \rangle$$

The vector's components, δg, δc and δt, are small discrete changes (measured in seconds) to *green* time, cycle length and *offset*, respectively. For example, if $\mathbf{s} = \langle 8, -4, 10 \rangle$, then the *green* time would be increased by 8 seconds, the cycle length reduced by 4 seconds and the *offset* increased by 10 seconds. In GRACE, auctions are executed periodically (line 19) as well. Every 5 minutes the auctioneer conducts an auction where the traffic signal agents bid against each other to win the authority to adjust the traffic signal timings at the intersection. As in SAT/Q, the auction mechanism has two parts but in GRACE, traffic signal agents perform an additional series of steps to determine their preferred adjustments to *split*, *cycle* and *offset* (line 20). However, the winner of the auction (line 21) is selected in the same way as it is done in SAT/Q (the traffic signal agent with the highest bid is chosen as the winner). The winner of the auction determines the changes that will be made to the traffic signal timing (line 22).

The auction process in GRACE is slightly different from the process used in SAT/Q. As mentioned above, in GRACE, traffic signal agents perform an additional procedure to find the most preferred adjustment, more specifically, the most preferred adjustment vector $\mathbf{s}$, where $\mathbf{s} \in S$. This new procedure is described in Algorithm 3. In GRACE, traffic signal agents evaluate the utility of every adjustment vector $\mathbf{s}$ in S (lines 7 to 13). The utility of an adjustment vector $\mathbf{s}$ is determined by a utility function:

$$U(\mathbf{s}) = S \to \mathbb{R}$$

Each traffic signal agent submits a preferred adjustment vector and bid to the auctioneer (lines 14 and 15). Two GRACE variants are implemented, MMDOS and DC2, their utility functions and bidding strategies are described in detail in Sections 4.4.1 and 4.5.

4.4.1 MMDOS

Presented here is MMDOS (*Minimise Maximum Degree Of Saturation*), a variant of GRACE named after the utility function used by its *traffic signal agents*. In MMDOS, *traffic signal agents* attempt to reduce the *degree of saturation* of the incoming link that is experiencing the highest level of use. This strategy is also used in SCOOT [26, 114].

In SAT/Q, saturation, the ratio of the traffic volume to its maximum capacity, is used to measure level of use. However, this ratio does not quantify how a change to *green* time (or *cycle*) effects the level of use on a link(s), hence, MMDOS relies on the *degree of saturation* [18, 115], X_i, which is defined as:

$$X_i = \frac{v_i L}{c_i g_i} \tag{4.3}$$

Eq. 4.3 gives the degree of saturation for the i^{th} group of incoming links at an intersection; recall, links are grouped together to form phases. Thus, v_i is the volume of traffic on a *critical* link (any link within the group given higher priority by traffic engineers [18]), while c_i, g_i and L are the maximum capacity for the critical link, the amount of *green* time allotted to the phase and the cycle length, respectively. Lastly, X_i is a value in the interval $[0, 1]$.

The utility of adjustment $\mathbf{s}$ in MMDOS is given by:

$$U(\mathbf{s}) = -[X(\mathbf{s}) + D(\mathbf{s})] \tag{4.4}$$

$$X(\mathbf{s}) = \frac{v_{max}(L + \delta c)}{c_{max}(g + \delta g)} \tag{4.5}$$

Eq. 4.5 gives the estimated degree of saturation if adjustment $\mathbf{s}$ is adopted. Traffic signal agents in MMDOS use the link (i.e., one of the links it manages) with the highest volume of traffic as its critical link. Thus, v_{max} is the volume of traffic on the link the traffic signal agent has designated as critical during the auction process. Likewise, c_{max} is the capacity of this same critical link. *Green* time, g, is the amount of *green* time allotted to the traffic signal agent and L is the cycle length at the intersection. Eq. 4.5 has the same range as Eq. 4.3, $[0, 1]$. In the utility function (Eq. 4.4), $D(\mathbf{s})$ is the estimated number of stopped vehicles[2] if adjustment $\mathbf{s}$ were adopted. The estimated number of stopped vehicles is calculated using historical traffic data from vehicle detectors and the method for counting stopped vehicles described in Section 4.2.1. Lastly, although in Eq. 4.4 the

[2]The value of $D(\mathbf{s})$ is scaled to $[0, 1]$

estimated number of stops is denoted as $D(\mathbf{s})$, only the *offset* component of vector $\mathbf{s}$ (or δt) is used in the estimation process.

The bidding rule[3] for MMDOS is:

$$bid = X_i \tag{4.6}$$

where X_i is the current degree of saturation for the links under the agent's control. In Eq. 4.6, the critical link is designated in the same manner as it is for Eq. 4.5 (i.e., the critical link is whichever link associated with the traffic signal agent which has the highest traffic volume).

The possible values for the components of vector $\mathbf{s}$, in MMDOS, are shown below:

$$\begin{aligned}
\delta g &\in \{0, 1, 2, 3, 4, 5\} \\
\delta c &\in \{-32, -16, -8, -4, 0, 4, 8, 16, 32\} \\
\delta t &\in \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}
\end{aligned}$$

4.4.2 MMDOS Variants

MMDOS adjusts all three traffic control parameters: *split*, *cycle* and *offset*. In order to examine the impact of the individual traffic control parameters on managing traffic flow at an intersection, six additional MMDOS variants are implemented. The variants represent the six combinations of *split* (S), *cycle* (C) and *offset* (O). The MMDOS variants are: MMDOS(C), MMDOS(O), MMDOS(OC), MMDOS(S), MMDOS(SC), and MMDOS(SO). The MMDOS variants function the same as MMDOS except they only adjust certain traffic control parameters, for example, MMDOS(C) will only adjust the *cycle* length, leaving the *offset* and *split* unchanged. The bidding function in the MMDOS variants remains the same as the bidding function for MMDOS, Eq. 4.6. However, the utility function is slightly different depending on whether or not the *offset* is adjusted.

$$U(\mathbf{s}) = \begin{cases} -[X(\mathbf{s})] & \delta t = 0 \\ -[X(\mathbf{s}) + D(\mathbf{s})] & \delta t \neq 0 \end{cases} \tag{4.7}$$

Lastly, the MMDOS variants have the same possible values for the components of vector $\mathbf{s}$ as the original MMDOS traffic control system.

[3]In SATQ, the bidding rule included a term (u_p) which represented the length of the vehicle queue on the link(s) assigned to the traffic signal agent. This term has been removed in order to stress the effects of the traffic control parameter, e.g., green time, on traffic performance.

Algorithm 2
GRACE: Traffic Control Algorithm.

```
t ← 0
H ← 15000                                          ▷ Maximum allowable time step.
S ← InitialiseAllPossibleAdjustments()             ▷ Finite set of adjustments.

J ← InitialiseIntersectionAgents()
for j ∈ Junctions do
    A_j ← InitialiseTrafficSignalAgents()
    T_j ← InitialiseTrafficSignalTiming()
end for

for j ∈ Junctions do
    for k ∈ A_j do
        Ŝ_k ← {s : s ∈ S, isAllowed(s)}    ▷ Variants are not allowed to perform all
possible adjustments.
    end for
end for

while true do
    for j ∈ J do
        if auctionTriggered() then                 ▷ Determined by auction period.
            P, B ← Auction(J)                      ▷ See Algorithm 3.
            k = determineWinner(B)                 ▷ Winning traffic signal agent
            updateSignalTiming(T_j, P, k)  ▷ Implement agent k preferred adjustment.
            setTrafficLights(T_j)
        else
            setTrafficLights(T_j)
        end if
    end for

    t ← t + 1
    if t > H then
        terminateSimulation()
    end if
end while
```

4.5 DC2: Dynamic Coalition Formation

SAT/Q and GRACE represent two market-based traffic control systems which utilise existing transportation technology for traffic control and acquiring data on traffic conditions. In SAT/Q, traffic signal agents only adjust the *split* or *green* time of traffic signal timings. In GRACE, traffic signal agents have greater control over traffic signal timing adjustment and can perform finer adjustments to the *split*, *cycle* and *offset*. However, in SAT/Q and in GRACE, traffic signal agents are only concerned with their respective traffic flows. That is, winning the auction is intended to improve the traffic flow associated with the winning traffic signal agent.

Algorithm 3
Auction Algorithm for GRACE.

```
procedure AUCTION(J)
    for j ∈ J do
        B ← ∅                                         ▷ Collection of bids
        P ← ∅                         ▷ Collection of preferred adjustments
        for k ∈ A_j do
            pIndex ← SystemMinInteger()
            for s ∈ Ŝ_k do
                i = U(s)          ▷ Utility function, see Sections 4.4.1 and 4.5
                if i > pIndex then
                    pIndex ← i
                    p_k = s       ▷ Traffic signal agent's preferred adjustment.
                end if
            end for
            P ← addAgentPreference(p_k)
            B ← submitBid(bid_k)  ▷ Bidding function, see Sections 4.4.1 and 4.5
        end for
    end for

    return P, B       ▷ Traffic signal agents' preferred adjustments and bids.
end procedure
```

This section presents DC2, a variant of GRACE where dynamic coalitions are formed to allow intersections to coordinate signal timing adjustments. In traditional coalitions, the temporary formation of agents is beneficial to all members of the coalition [116]. Although the coalitions that are described in my work are also temporary, not all agents within the coalition will directly benefit from the formation. In this thesis, a coalition is a temporary pairing of intersections for coordination via *offset* adjustments where the preferred *offset* adjustment of a traffic signal agent is subject to upstream traffic conditions. That is, in my coalitions, one intersection temporarily utilises traffic stream information from another intersection to guide *offset* adjustments. It is worth noting though, that the upstream intersection, i.e., the intersection which provides traffic stream information, indirectly benefits from the coalition in that the link shared by both intersections within the coalition is less likely to suffer from spillback.

Sandholm *et al.* [117] identified three phases of coalition formation: *coalition structure generation* (determining coalition membership as to maximise the total value of all coalitions), *solving the optimisation problem* (assigning a task(s) to each member such that every coalition member benefits from membership, i.e., each member is better off helping by taking part in solving the joint problem of the coalition given its resources) and *dividing the value* (distributing the value of the utility to its members, in other words, ensuring that each member of the coalition has something to gain from being part of a coalition). Finding the optimal coalition is NP-complete [117]; each phase of the coalition formation process has its own unique challenges. However, these challenges are lessened within the traffic domain, as well as within my approach for determining signal timing adjustments. First, the difficulty in determining ideal coalition membership is diminished because *offset* adjustments are made in relation to neighbouring intersections. Similarly, issues that may arise in *solving the optimisation problem* and *dividing the value* are no longer an issue given the manner in which coordination is facilitated between the two intersections within a coalition.

SCOOT employs *offset* adjustments for intersection coordination as well. However, in SCOOT the intersections that are coordinated are fixed radial (linear path) road networks. In DC2 the coalitions are dynamically formed and dissolved once they are no longer of value. Additionally, using *offset* adjustments for coordination sets DC2 apart from other market-based traffic systems. In many market-based traffic systems traffic signals are either removed from the traffic control system, e.g., [64, 67] or *offset* adjustments are ignored [63, 71].

DC2 is different from MMDOS (and the other MMDOS variants which adjust *offset*) in that the *offset* that is adopted may not have been preferred by the traffic signal agent that won the auction. In DC2, $D^I(\mathbf{s})$ returns the estimated number of stopped vehicles if adjustment $\mathbf{s}$ were adopted but in relation to the intersection, that is, if the intersection were allowed to apply the *offset* in any manner it chooses. The utility of adjustment $\mathbf{s}$ in DC2 is given by:

$$U(\mathbf{s}) = -[X(\mathbf{s}) + D^I(\mathbf{s})] \tag{4.8}$$

The term $D^I(\mathbf{s})$ is used here instead of $D(\mathbf{s})$ to denote that it returns the estimated number of stopped vehicles in relation to the intersection, as explained above. The bidding rule for DC2 is:

$$bid = X_i \tag{4.9}$$

where X_i is the current degree of saturation for the links under the agent's control (as described for MMDOS traffic signal agents in Section 4.4.1). Other than the difference between $D(\mathbf{s})$ and $D^I(\mathbf{s})$, the utility function and bidding rule for DC2 works as it does in MMDOS ($X(\mathbf{s})$ and X_i are described in Section 4.4.1).

The possible values for the components of vector $\mathbf{s}$, in DC2, are shown below:

$$\begin{aligned}
&\delta g \in \{0, 1, 2, 3, 4, 5\}\\
&\delta c = 0\\
&\delta t \in \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}
\end{aligned}$$

DC2 does not change the cycle length; this is a requirement needed to form green waves. Thus, δc is zero.

Figure 4.5 shows a sample set of coalitions that were formed during a test simulation, each frame is a snapshot of the coalition formed during a cycle. In Figure 4.5, each blue circle represents an intersection and points to the other member of the coalition. Arrows are pointing towards an upstream intersection, i.e., the target traffic stream is moving in the opposite direction of the arrow.

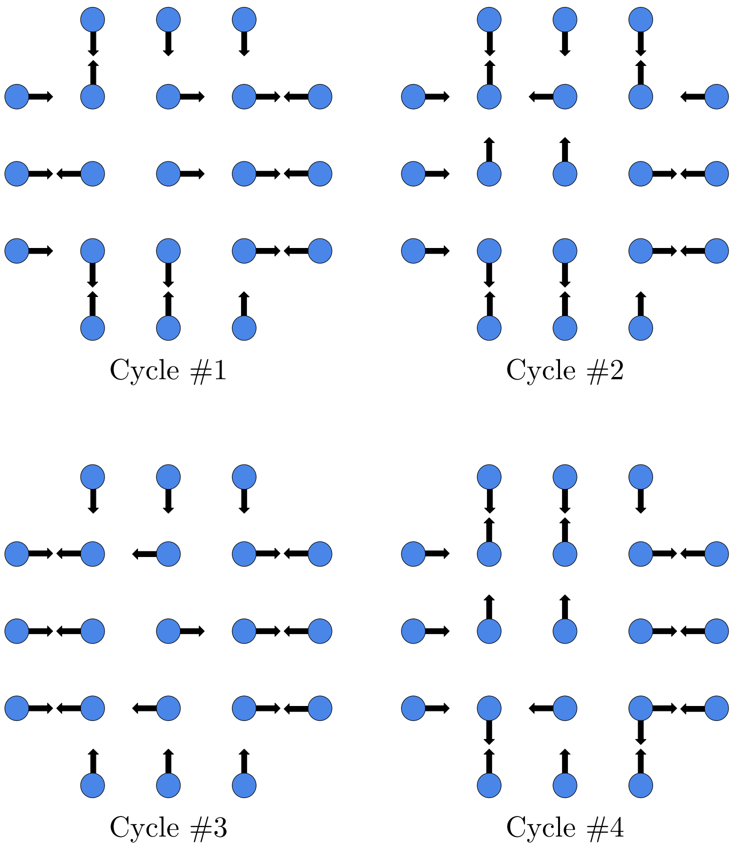


FIGURE 4.5: Snapshots of coalitions that are formed under DC2 during a test simulation run. Each blue circle represents an intersection and the arrows show the intersection's partner in the coalition. More specifically, the arrows point to the source of the traffic stream that the downstream intersection will try to improve using *offset* adjustments.

4.6 Summary

This chapter described four market-based traffic control systems, SAT, SATQ, MMDOS and DC2, that are designed to be deployed using existing transportation infrastructure. Additionally, six MMDOS variants, which adjust different combinations of *split*, *cycle*, and *offset*, are proposed. Table 4.1 list all of the market-based mechanisms presented in this chapter and the traffic control parameters that they adjust.

	Traffic Control Parameters		
Mechanism	*Split*	*Cycle*	*Offset*
SAT	*Periodically*	-	-
SATQ	*Periodically*	-	-
MMDOS	*Periodically*	*Periodically*	*Periodically*
MMDOS(S)	*Periodically*	-	-
MMDOS(C)	-	*Periodically*	-
MMDOS(O)	-	-	*Periodically*
MMDOS(OC)	-	*Periodically*	*Periodically*
MMDOS(SC)	*Periodically*	*Periodically*	-
MMDOS(SO)	*Periodically*	-	*Periodically*
DC2	*Periodically*	-	*Periodically*

TABLE 4.1: Market-based traffic control systems. Traffic control parameters are labelled *Periodically* because the auctions, which result in traffic signal timing changes, occur periodically.

Chapter 5

Evaluation of Market-based Systems

5.1 Introduction

This chapter describes the experimental environment and performance metrics used to evaluate the multi-agent market-based traffic control systems developed for this thesis. Additionally, this chapter includes a description of the traffic control mechanisms that are used as benchmarks. The experimental environment is composed of the traffic simulator (SUMO), road networks and traffic conditions. The chapter begins with a description of the traffic simulator in Section 5.2. Section 5.3 describes the road networks that are used in the traffic simulator. Section 5.3 also discusses the formation of gridlock which is tied to traffic generation in SUMO. Section 5.4 defines the set of traffic conditions that are used to evaluate performance. Sections 5.5, 5.6 and 5.7 describe the traffic control methods that are used as benchmarks. Lastly, the performance metrics and method of statistical analysis are explained in Sections 5.8 and 5.9.

5.2 Experimental Environment: SUMO

There is strong support for the use of traffic simulators for the testing of traffic control systems prior to their deployment in the field [24]. Traffic simulations are a vital first step in evaluating traffic control mechanisms to answer many questions about the viability of a proposed traffic control system at a fraction of the cost of running real-world tests. Additionally, there are safety issues that must be considered when testing traffic signal control algorithms, that is, poorly timed traffic signals can cause accidents. Simulations are a safe and economical way of evaluating traffic control and assignment strategies. Simulation studies in the traffic domain are done on a variety of traffic simulations, e.g., PARAMICS, VISSIM, DRACULA, and CORSIM, there is no one standard traffic simulator.

Simulation of Urban MObility (SUMO) [118] is an open-source microscopic traffic simulator developed by DLR (Institute of Transportation Systems). SUMO is designed to address a large set of traffic management topics. As a microscopic traffic simulator, each vehicle is modelled explicitly, i.e., each vehicle has its own route and moves independently through the network. SUMO can run with or without a GUI front end. Some of its key features include: space-continuous and time-discrete vehicle movement, different vehicle types, multi-lane streets with lane changing, different right-of-way rules, traffic lights, network-wide, edge-based, vehicle-based, and detector-based outputs. SUMO has been used to evaluate traffic signal control algorithms, navigation systems (route choice and re-routing algorithms), and $V2V$ and $V2I$ communications.

SUMO was designed to allow traffic researchers to evaluate their traffic control algorithms without having to worry about the underlying traffic simulation components such as the road network and traffic signals. SUMO's Traffic Control Interface (TraCI) allows traffic researchers to retrieve simulation state data and manipulate the behaviour of simulated objects. This thesis utilises a Python version of TraCI to implement the market-based traffic control systems and the traffic control systems used as benchmarks.

Traffic simulation models rely on a large number of parameters to determine how the traffic model will run. Simulation parameters are used to configure the behaviour and characteristics of the simulation environment, i.e., driver route choice, driver behaviour and road network conditions. SUMO has two key steps in preparing a simulation scenario: building the road network and preparing traffic demand. The road networks are described in Section 5.3 and the traffic demand in Section 5.4. Lastly, the SUMO agent driver model is described below in Section 5.2.1.

5.2.1 Driving Model

Only passenger vehicles are used in my experiments, i.e., the traffic simulations do not include buses, cyclists or pedestrians. Vehicle agents in SUMO are rational agents with a single goal: reach their destination as quickly and as safely as possible. In SUMO, vehicle agents employ a *car-following* model to simulate driving behaviour. The default *car-following* model in SUMO (and the one utilised in my work) was developed by Stefan Krauss [119]. Vehicles maintain the fastest and safest possible driving speed when using Krauss' *car-following* model. More specifically, the vehicle's speed will not exceed the vehicle's maximum allowable speed [1] and the vehicle's speed will always allow the vehicle to safely react to changes in the speed of the lead vehicle (i.e., the vehicle being followed if one exist). Additionally, vehicles only perform legal manoeuvres including not entering the intersection box unless there is ample room to pass completely through it. Table 5.1 is a complete list of the *car-following* parameters (and their values) used in my simulations.

[1]The maximum vehicle speed and other parameters used by the *car-following* model are shown in Table 5.1

Parameter	*Value*
Acceleration	$0.8m/s^2$
Deceleration	$4.5m/s^2$
Sigma (driver imperfection)	0.5
Length of vehicle	$5m$
Minimum gap	$2.5m$
Maximum speed	$16.67m/s$

TABLE 5.1: Driver agent parameters. Sigma (a value between 0 and 1) produces stochastic driving behaviour. Any non-zero value for sigma introduces small random changes to the vehicle's speed. The minimum gap is the closest a vehicle will approach the lead vehicle (i.e., the vehicle it is following).

5.3 Road Networks

Modern urban planners balance the need for safe pedestrian movements against vehicle movements [2] amongst other things. The road networks of many cities reflect local desires or priorities set by governing bodies on traffic issues. Simply adopting the topology of an existing city ignores countless aspects, e.g., geography and local history, that shaped the city's road network but have little to do with transportation efficiency. Large road networks that develop organically, i.e., those developed over time with little fore-planning, are difficult to recreate in simulation. To facilitate the study of the traffic control systems described in this thesis, four key road network properties were identified:

i. Easy to create in simulations

ii. Capable of recreating the complexity found in real world road networks

iii. Identifiable base elements that can be classified

iv. Ability to use *green waves* as a form of coordination amongst intersections (So, the road topology should be open to scrutiny in order to search for relations between road geometry, intersection placement and intersection coordination).

Grid-based road networks were utilised to evaluate the traffic control systems described in this thesis. A key advantage of using grid-based plans is that they are more reproducible in other traffic simulation software than road networks that are formed organically, e.g., London or Paris. In grid-based networks, streets are orthogonal to one another. The order and structure of the grid plan model is ideal for the study of intersection coordination. Other researchers that have used orthogonal streets and/or road networks based on the grid plan include [47, 51, 86, 99, 120].

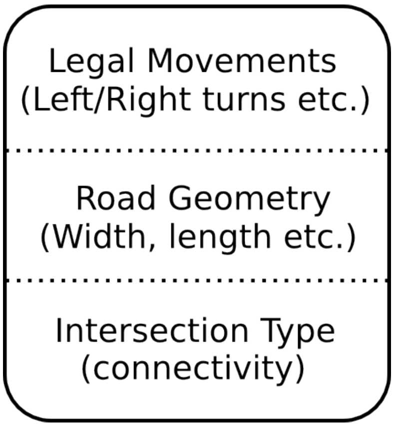


FIGURE 5.1: Layered components of a road network.

A simulated grid-based road network can be described as a 3-tiered structure, shown in Figure 5.1. At the very bottom are intersection types, then road geometry, followed by legal vehicle movements. Figure 5.2 illustrates the four types of intersections that have been identified: T-junction, L-junction, four-way junction and the cul-de-sac. In road networks where only four-way junctions are used a new structure is formed, the *block*. A *block* is the area in a grid plan enclosed by streets. The *block* can be used to describe different types of grid-plan road networks based on the ratio of the width to length of the *block*. The grid-plan model allows us to build realistic road networks with a well-defined palette of road elements. The road networks utilised in this work can be described with ease for either reconstruction in another traffic simulator or for more in-depth study.

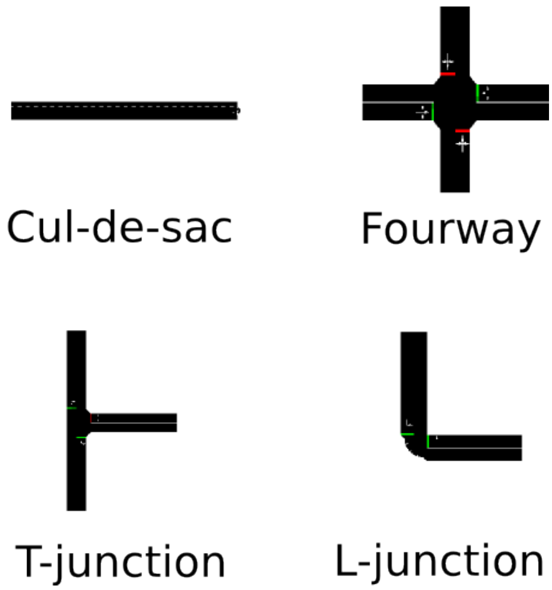


FIGURE 5.2: Intersection types found in grid plan model.

Grid-based road networks can control the balance between the needs of pedestrians and vehicle movements by adding or removing certain roadway features. Cul-de-sacs, T-junctions and hierarchical streets (e.g., one-way streets versus a four lane two way street) are often used as a way to dampen traffic in an area [2]. These elements are used to produce a network that is more suited for foot traffic. For example, Traditional Neighbourhood Design (TND) is exactly such an idea; it is a grid-based network which includes T-junctions and one-way streets used to lessen traffic [2]. Grid-based networks that include traffic abatement measures such as the ones mentioned above are called Fused Grids [2], illustrated in Figure 5.3.

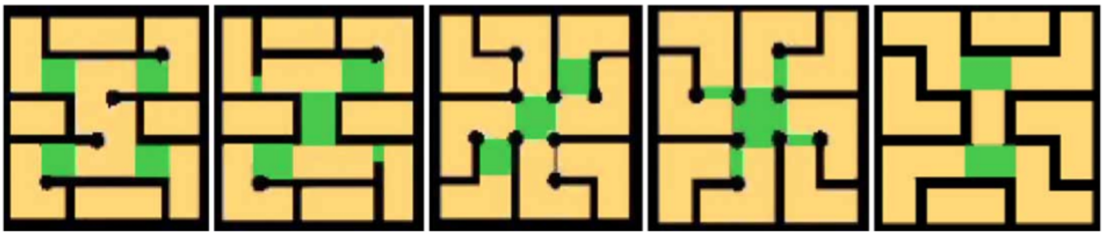
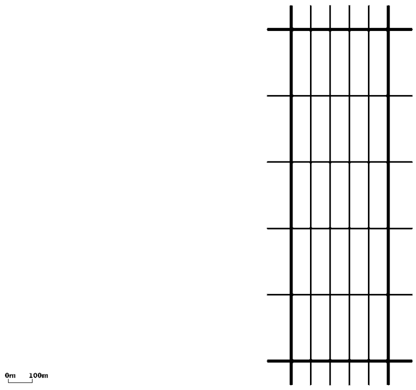
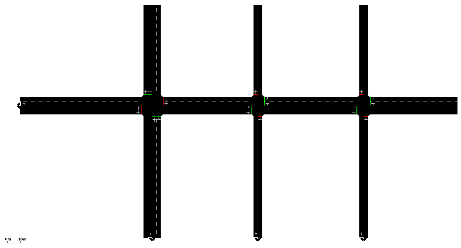


FIGURE 5.3: Five examples of Fused Grid neighbourhoods. The green squares (and rectangles) represent green spaces, e.g., parks and playgrounds [2].

5.3.1 Portland Road Network



(a) Portland road network.



(b) Close up of two-lane avenue.

FIGURE 5.4: Portland road network on SUMO.

The Portland road network, shown in Figure 5.4a, is modelled after the Portland, OR (USA) grid plan. The Portland map has two-way, single lane, streets running East/West and North/South. However, it also has a large avenue running along its perimeter which is two-way with four lanes. The distance between horizontal traffic signals (East/West), is $80m$ and between vertical traffic signals (North/South) is $280m$. The Portland map covers 0.96 km^2 and has 36 traffic signals. Traffic signals use a two-phase signal plan: during one phase, north/south bound traffic passes through the intersection, while west/east bound traffic passes in the other phase. Additionally, roadways do not include dedicated turning (right or left) phases; therefore left and right turns were given lower priority than through movements, i.e., vehicles turning left or right waited until it was safe to do so.

5.3.2 Phoenix Road Network

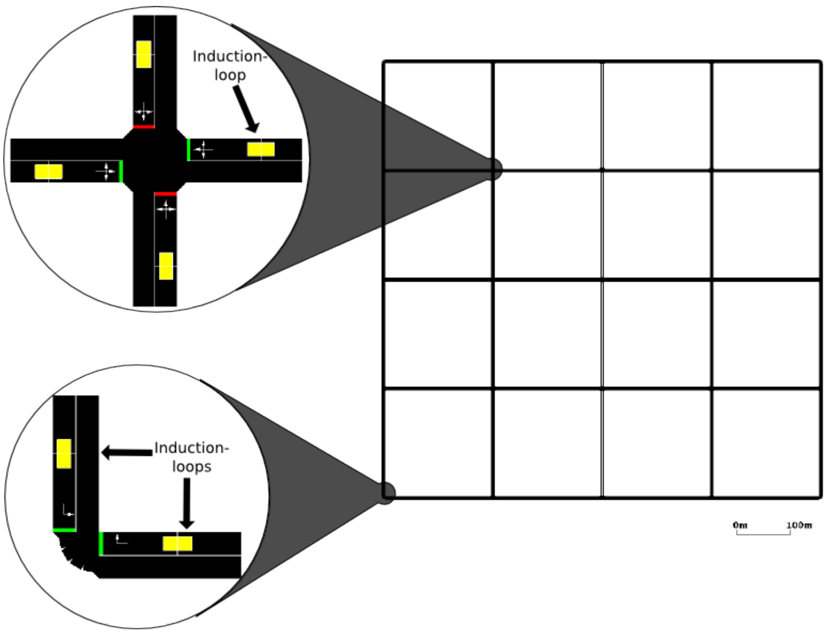


Figure 5.5: Phoenix road network on SUMO.

The Phoenix road network is a 5×5 grid-based city (Figure 5.5). The simulated city is organised in square *blocks* where the distance from one intersection to the next is 200 meters. The Phoenix map covers 0.64 km^2 and has twenty-one traffic signals. The four traffic signals in the corners of the network are deactivated because these four traffic signals control streams run without conflicts (i.e., vehicles traversing these intersections will never have to yield to one another). The roadways on the Phoenix map are all one ways and do not have a dedicated turning lane. All traffic signals use a two-phase signal plan: during one phase, north/south bound traffic passes through the intersection, while west/east bound traffic passes in the other phase. The signal plan did not include dedicated turning (right or left) phases; therefore left and right turns were given lower priority than through movements.

5.4 Traffic Conditions

The market-based traffic control systems developed in this thesis are evaluated in six traffic scenarios. The traffic scenarios represent traffic patterns that are typically found within SCOOT *regions* as well as those that are not. The scenarios included disruptions to traffic flow in the form of sudden increases in traffic intensity. Two characteristics help categorise the traffic scenarios: traffic demand for each traffic signal phase and variations in traffic demand between connected intersections. Hence, the traffic scenarios can be categorised as either *predictable* or *unpredictable*. In *predictable* traffic scenarios, the traffic demand on North/South bound traffic is heavier than East/West bound traffic. Additionally, in *predictable* traffic, the ratio of North/South traffic demand to East/West traffic demand remains the same on connected intersections, i.e., the direction of the heaviest flow remains the same. However, in *unpredictable* the ratio of North/South traffic demand to East/West traffic demand is different from one intersection to another.

The *predictable* traffic scenarios are:

- **Structured**—a traffic flow through the network with an identifiable (e.g., commuter) path with heavy volume. *Structured* traffic is ideal for UTCs such as SCOOT. If the heavy traffic flows have been identified and are predictable, then those intersections with the heavy traffic flow can be assigned to a SCOOT *region*. In *structured* traffic, a disruption occurs at the one-hour mark (3600^{th} second) and lasts for one hour. During the disruption, traffic intensity is increased by 18%.

- **Regional** is identical to *structured*, except that cross traffic is kept at minimal levels.

- **Constant** traffic flow has a static volume of traffic entering the network. Traffic demands remains the same during the entire scenario. Also, the traffic demand on opposing traffic signal phases is identical. The traffic demand used in the *constant* traffic scenario is the same as the scenario described in [32].

The *unpredictable* traffic scenarios are:

- **Unstructured**—a traffic flow with no identifiable path with heavy volume. Without prior knowledge of traffic flows, *unstructured* traffic poses a challenge to SCOOT-like systems. Along with randomly chosen paths, *unstructured* traffic has a disruption at the one-hour mark (3600^{th} second), again it lasts for one hour, during which traffic intensity is raised by 55%.

- **Football**—a traffic flow that emulates road conditions before, during and after a special event like a football match. This scenario represents a worst-case scenario where there are two sudden and sharp increases in traffic demand. The first disruption occurs when football fans enter the area of an arena (30 minutes after the simulation started); and second, when fans exit the arena (approximately 90 minutes later). Additionally, unlike in the *structured* and *unstructured* traffic scenarios where trips ended outside of the city, the final destination of vehicles during the first disruption is within the city limits (i.e., at the arena location).

- **Directional** is similar to *structured*, but there is a shift in the direction of the heavy flow midway through each experiment.

5.5 SCOOT

The market-based traffic control mechanisms developed in this thesis (and the SCOOT variants) are compared to three other traffic control systems: SCOOT [26] (described in this section), a reinforcement-learning based traffic control system (SUPRL) described by Bazzan *et al.* [32] (Section 5.6) and fixed-time traffic signals (Section 5.7).

Prior to the popularity of adaptive traffic control systems, traffic engineers would develop fixed-timed signal plans for different traffic conditions or events [26]. For example, traffic engineers would have a fixed-time plan specifically for morning or evening rush hour. Signal plans can also be designed for holidays or major events, e.g., football matches [26]. Traffic engineers used historical traffic flow data to develop the signal plans for an area, usually a small subset of intersections within a larger city network. However, there were major drawbacks to using historical traffic flow data; mainly the cost in time and money to collect the accurate data. In-ground vehicle detectors were not readily available, thus, traffic engineers actually had to go out in the field to collect traffic flow measurements. Many cities habitually used old signal timing plans because collecting new traffic flow data was so cost prohibitive [26]. Furthermore, the fixed-time signal plans were prone to errors. The plans did not fare well when traffic flow deviated from expected levels, during accidents or during any other unexpected disruptions to traffic flow. Additionally, once the data was collected it was still difficult to manually determine optimal signal timing. Applications, such as TRANSYT [26, 101] were developed for this purpose, to calculate fixed-time signal plans to minimise delay (or some other traffic metric such as vehicles stops or fuel consumption) [26]. Transport Research Laboratory (TRL) developed TRANSYT to make the design of fixed-time signal plans easier, however, it worked offline and relied on historical traffic flow data. Thus, it was subject to the same error conditions as the older manually developed plans.

As the availability of real-time traffic flow data grew, so did the abilities of traffic control systems. Instead of feeding traffic control systems historical data newer applications were developed that could select an appropriate fixed-time signal plan given current traffic conditions. SCOOT (Split, Cycle and Offset Optimisation Technique), also developed by TRL, is a dynamic and real-time adaptive traffic control system. SCOOT addresses two major issues with the older systems such as TRANSYT: outdated traffic flow data and fixed-time signal plans. SCOOT minimizes delay and prevents congestion by coordinating traffic signals in real time from a centralized computer. It is used in over 14 countries around the world [3].

After over three decades of use, SCOOT continues to evolve in order to better manage traffic. Although SCOOT still works on the same principles originally described in [26], it has become a more rounded and comprehensive UTC capable of prioritising transit buses, gating behaviours and incorporating pedestrian crossing. To improve travel times of transit buses (and other vehicles considered high priority), newer versions of SCOOT can skip phases. That is, SCOOT can skip to the phase that would provide the best travel times for bus(es) on a specified route. When gating, SCOOT will restrict the flow of vehicles entering an area, thus, shifting vehicle queues to lower priority road segment(s). Lower priority road segments are any areas (designated by traffic engineers) of less concern where vehicle queues can form without harming critical junctions [121]. In other words, gating allows vehicle queues to be relocated to other areas of a road network. Lastly, newer versions of SCOOT can take advantage of the variable length of the pedestrian phase (this is the traffic signal phase where pedestrians are allowed to cross the intersection) to further improve traffic flow.

For this thesis, a SCOOT emulator was developed for SUMO. The reason behind this decision is twofold: (1) to analyse the use of traffic control parameters, e.g., adjusting *split* versus adjusting the *cycle* length, in SCOOT; and (2) to provide the ability to replicate elements of SCOOT within other (market-based) traffic control systems.

The CORSIM (Corridor Simulation) traffic simulator has a SCOOT interface [122]. The SCOOT-CORSIM interface allows SCOOT to receive traffic data from the simulation in CORSIM and send signal timing updates to the traffic signals in the simulation. However, this assumes the user also has access to the SCOOT software. Additionally, CORSIM is a traffic simulator with strong ties to Federal Highway Administration (FHWA). FHWA is an agency within the US Department of Transportation responsible for the construction and maintenance of US's highways, bridges and tunnels in the US. Thus, CORSIM is far more popular in the US than it is in other countries. Furthermore, both CORSIM and SCOOT are not free software. However, the principles with which SCOOT optimises signal timings are well documented [26, 101, 114, 123]

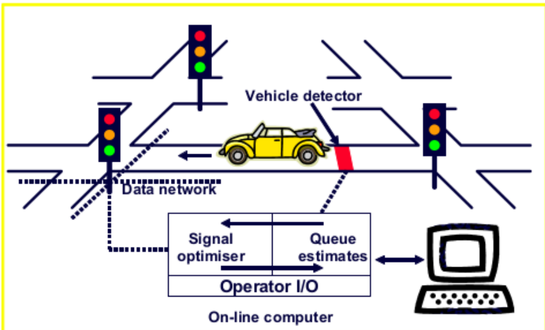


Figure 5.6: SCOOT centralized signal optimiser [3]

SCOOT relies on in-ground vehicle detectors at each source intersection. The detectors are used by SCOOT to form a model of the traffic flow on the street. Figure 5.6 illustrates the relationship between the vehicle detectors, signal to be optimised and the central computer. Data from the vehicle detectors are sent to a central computer where it is used to model traffic flow at an intersection downstream from the detector.

Algorithm 4
SCOOT Traffic Control Algorithm.

```
1:  t ← 0
2:  H ← 15000                                         ▷ Maximum allowable time step.
3:  trafficFlow ← 0
4:
5:  J ← InitialiseSCOOTIntersections()
6:  R ← InitialiseSCOOTRegions()
7:  while true do
8:      for j ∈ J do
9:          if t mod 4 = 0 then
10:             trafficFlow ← estimateDownStreamVolume()
11:         end if
12:
13:         if isShowingGreen() and remainingGreenTime = 5 then
14:             AdjustSplit(j, trafficFlow)                ▷ See Algorithm 5.
15:         end if
16:
17:         if endOfCycle() then
18:             AdjustOffset(j, trafficFlow)               ▷ See Algorithm 7.
19:         end if
20:     end for
21:
22:     if updateRegion() then        ▷ updateRegion() returns true every 5 minutes
23:         for r ∈ R do
24:             AdjustCycle(r)                             ▷ See Algorithm 6.
25:         end for
26:     end if
27:
28:     t ← t + 1
29:     if t > H then
30:         terminateSimulation()
31:     end if
32: end while
```

The general SCOOT procedure is described in Algorithm 4 with more details on how SCOOT adjusts *split*, *cycle* and *offset* in Sections 5.5.1, 5.5.2 and 5.5.3. SCOOT coordinates the traffic signal timings of small sets of intersections called *regions*. Regions are pre-defined (line 6) by traffic engineers and do not change during the execution of SCOOT. All the intersections within a *region* share the same cycle length. SCOOT *regions* form linear paths, i.e., a series of intersections without any turns. SCOOT uses small incremental changes to *split*, *cycle* and *offset*, to improve traffic signal timing. Adjustments to the three parameters are made at different times during the cycle using the degree of saturation of incoming links, an estimation of stopped vehicles and a model of downstream traffic volume (the traffic model is updated every four seconds, lines 9 to 11). The *split* is updated five seconds before a phase ends (lines 13 to 15). Within a *region*, the *split* is adjusted to reduce the degree of saturation of the incoming link with the highest level of use. The cycle length is also adjusted to maintain optimal levels of traffic demand; cycle length is adjusted every five minutes (lines 22 to 26). SCOOT tries to adjust the cycle length to maintain a maximum *degree of saturation* near 90% (i.e., at the intersection level). Lastly, within a *region*, adjustments to the *offset* are made to reduce the number of stops (lines 17 to 19). The *offset* is adjusted at the end of every cycle.

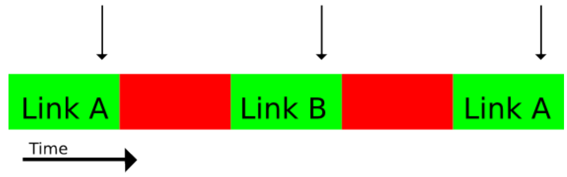


FIGURE 5.7: *Split* adjustment frequency. The green segments represent the time period when a link(s) has the *green* light while the red segments represent a transitional period (usually an amber light followed by a red light).

5.5.1 Split

A traffic signal cycle is composed of phases and each is allotted green time. The division of green time, amongst the phases, is called the *split*. The *split* optimiser tries to reduce the maximum *degree of saturation* on links approaching an intersection. SCOOT's procedure for updating the *split* is described in Algorithm 5. The *split* optimiser makes both *temporary* and *permanent* signal timing changes. Temporary changes (in increments/decrements of 4 seconds) are in place until the cycle ends and permanent changes (in increments/decrements of 1 second) are seen in subsequent phases (lines 2 and 3) [114]. Figure 5.7 shows the timing of *split* adjustments within a traffic signal phase. Five seconds before a phase change, SCOOT considers the effect on the *degree of saturation* caused by *advancing* (terminating the phase), *retarding* (extending the phase) or *holding* (allowing the phase to continue to termination), lines 5 to 7. In other words, SCOOT estimates the degree of saturation for each case and selects the option that reduces the degree of saturation the most (lines 8 to 16).

Each incoming roadway(s) at an intersection has its own *degree of saturation*, assuming there is a vehicle detector for that roadway(s). The *degree of saturation* is shown in Eq. 5.1 [115]. In Eq. 5.1, $V(t+T)$ is the volume of traffic SCOOT predicts will arrive at the intersection from another upstream intersection in the *region*. SCOOT uses the same method, Eq. (5.2), as TRANSYT to predict this downstream traffic volume [1], i.e., the volume of traffic downstream from a vehicle detector. Table 5.2 provides additional details on the terms used to estimate the downstream flow and *degree of saturation* in SCOOT. Note, the degree of saturation in my auction-based traffic control system is the same as it is in SCOOT —a measurement of the level of use of a roadway. However, the difference between the two lies in how SCOOT measures traffic volume and designates critical links.

Algorithm 5
SCOOT's Algorithm for updating the *split*.

```
procedure AdjustSplit(j, trafficFlow)
    permanentSplitChange ← 4
    temporarySplitChange ← 1
        ▷ checkDos() uses Eq. 5.1 to estimate the degree of saturation if the split is changed.
    keep = checkDos(0, trafficFlow)                          ▷ Keep the current split
    retard = checkDos(splitChange, trafficFlow)              ▷ Extend green time.
    advance = checkDos(−1 × splitChange)                     ▷ Shorten green time.
    r = minIndex([keep, retard, advance])

    if r = 2 then
        updateSplit(T_j, permanentSplitChange, temporarySplitChange)
    end if

    if r = 3 then
        updateSplit(T_j, −1×permanentSplitChange, −1×temporarySplitChange)
    end if
end procedure
```

First, SCOOT uses TRANSYT's downstream traffic model to predict traffic volume at the intersection while in my approach, traffic volume is determined by the amount of upstream traffic. Second, critical links are pre-defined in SCOOT (the purpose of SCOOT is to improve traffic flow along a linear path). In my approach, the critical link (of a phase or traffic signal agent) may change during execution of the traffic controller; it depends on which link is experiencing the highest level of use.

$$X(t) = \frac{V(t+T)L}{cg} \tag{5.1}$$

$$V(t+T) = F \times V(t) + [(1-F) \times V(t+T-1)] \tag{5.2}$$

Variable	*Value*
$V(t)$	Predicted flow rate (at time t) of the platoon
T	0.8 times the cruise travel time on the roadway
F	A smoothing factor where $\left(F = \frac{1}{1+aT}\right)$ and a is the platoon dispersion factor[a]
c	The maximum possible volume of traffic on the link
g	*Green* time
L	*Cycle* length

Table 5.2: Prediction of down stream flow rate and *degree of saturation* variables (used by TRANSYT and SCOOT) [1].

[a] Gordon and Tighe [1] suggests using 0.5 (high traffic demand), 0.35 (moderate traffic demand), or 0.25 (low traffic demand) depending on traffic conditions.

5.5.2 Cycle

Cycle length is the total amount of time it takes for every phase to receive its portion of green time. SCOOT's procedure for updating the cycle length is described in Algorithm 6. The *cycle* optimiser runs every 5 minutes, but can run anywhere from every 2.5 to 10 minutes. SCOOT optimises the cycle length by examining the *degree of saturation* of all the intersections in the *region* (line 2). The intersection with the highest saturation level is considered the *critical node* [114] —this critical node determines how the cycle lengths of all the intersections in the *region* will be changed. Cycle length changes are made in increments/decrements of 4, 8, 16 and 32 seconds; the shorter the cycle, the smaller the change (line 4) [114, 124]. If the highest degree of saturation is greater than 90%, then the cycle length (for the entire *region*) is increased; otherwise SCOOT decreases the cycle length (lines 6 to 16). The lower bound for cycle length, normally $30 - 40$ seconds, is limited by parameters such as pedestrian crossing time and minimum green time [114]. The upper bound is normally $90 - 120$ seconds [114].

Algorithm 6
SCOOT's Algorithm for updating the *cycle* length.

1: **procedure** ADJUSTCYCLE(J_r) ▷ J_r is the set of junctions that belongs to a region
2:     $d \leftarrow maximumInterseciontDOS(J_r)$
3:
4:     $cycleChange \leftarrow getCycleChange()$ ▷ The change in cycle length depends on the current cycle length, See Section 5.5.2
5:
6:     **if** $d < 0.9$ **then**
7:         **for** $j \in J_r$ **do**
8:             $updateCycle(T_j, -1 \times cycleChange)$
9:         **end for**
10:     **end if**
11:
12:     **if** $d >= 0.9$ **then**
13:         **for** $j \in J_r$ **do**
14:             $updateCycle(T_j, cycleChange)$
15:         **end for**
16:     **end if**
17: **end procedure**

5.5.3 Offset

A *green wave* is a phenomenon that occurs when a platoon (cluster of vehicles travelling at similar speeds) crosses a series of intersections with very few stops. The *offset* parameter is a central component in the formation of green waves and thus, traffic signal coordination. In order for a green wave to occur, the traffic signals at adjacent intersections in a given path must have the same cycle length. The *offset* parameter represents the difference between the start of cycles at two consecutive intersections. SCOOT's procedure for adjusting the *offset* is described in Algorithm 7. SCOOT checks the *offset* once at the end of every cycle. *Offset* adjustments are made in increments/decrements of 4 seconds (line 1) [114]. SCOOT considers the effect on the estimated number of stops caused by keeping, increasing or decreasing the *offset* (lines 4 to 7). SCOOT selects the option that produces the greatest reduction in the estimated number of stops (lines 7 to 15). SCOOT estimates the number of stopped vehicles using historical traffic data and the method described in Section 4.2.1.

Algorithm 7
SCOOT's Algorithm for updating the *offset*.

1: $\mathit{offsetChange} \leftarrow 4$
2:
3: **procedure** ADJUSTOFFSET($j, \mathit{trafficFlow}$)
▷ *checkOffset*() uses the method described in Section 4.2.1 to estimate the number of stops if the offset is adopted.
4: $\quad \mathit{keep} = \mathit{checkOffset}(0)$
5: $\quad \mathit{increase} = \mathit{checkOffset}(\mathit{offsetChange})$
6: $\quad \mathit{decrease} = \mathit{checkOffset}(-1 \times \mathit{offsetChange})$
7: $\quad r = \mathit{minIndex}([\mathit{keep}, \mathit{increase}, \mathit{decrease}])$
8:
9: $\quad$ **if** $r = 2$ **then**
10: $\qquad \mathit{updateOffset}(T_j, \mathit{offsetChange})$
11: $\quad$ **end if**
12:
13: $\quad$ **if** $r = 3$ **then**
14: $\qquad \mathit{updateOffset}(T_j, -1 \times \mathit{offsetChange})$
15: $\quad$ **end if**
16: **end procedure**

5.5.4 SCOOT Variants

Normally, SCOOT adjusts all three traffic control parameters –*split*, *cycle* and *offset*. In order to examine the impact of the individual traffic control parameters on managing traffic flow at an intersection, six additional SCOOT variants are implemented. The variants represent the six combinations of *split* (S), *cycle* (C) and *offset* (O). The SCOOT variants are: SCOOT(C), SCOOT(O), SCOOT(OC), SCOOT(S), SCOOT(SC), and SCOOT(SO). The SCOOT variants function the same as SCOOT except they only adjust certain traffic control parameters, for example, SCOOT(C) will only adjust the *cycle* length, leaving the *offset* (the default *offset* is zero) and *split* unchanged.

5.6 Reinforcement-Learning Based Traffic Controller

In addition, the reinforcement-learning traffic control system (SUPRL) described by Bazzan *et al.* [32] is implemented on the SUMO testbed. Bazzan *et al.* [32] utilised *supervisors* as high-level agents that observe small groups of intersection-level agents (or *local* agents) in search of an optimal joint policy. In [32], the subordinate agents (intersections) could perform three actions; each action was a complete traffic signal timing plan. The SUMO implementation of SUPRL has three actions as well. The first action is a neutral traffic signal setting, i.e., the green time is evenly divided amongst the two phases (exact values are shown in Table 5.3). The second action favours the phase that services the north/south-bound lanes (by allotting more green time to the phase), in this plan 50 seconds is allotted to north/south-bound traffic and 20 seconds to west/east-bound. Lastly, the third action favours the west/east-bound lanes, in this plan 20 seconds is allotted to north/south-bound traffic and 50 seconds to west/east-bound. The Bazzan *et al.* [32] traffic control system shares some characteristics with SCOOT. The most notable similarity is that both systems work within small groups of intersections. Therefore, in the simulation experiments, the *supervisors* and subordinates are organised in the same configuration as the SCOOT *regions*. Three *supervisors* are used, each with three subordinate agents arranged in a north-to-south configuration. Also, local agents can decide not to perform the action suggested by the supervisors. More details on SUPRL can be found in Section 3.6.

5.7 Fixed-Time Traffic Signals

For empirical evaluation, a fixed-time traffic signal controller that represents traditional (non-adaptive) traffic signal devices is also implemented on the SUMO testbed. These traffic signals display the same light sequences for the same duration every cycle. The traffic signal timing used by the fixed-time traffic signals are the same initial traffic signal timing used by the adaptive mechanisms. Thus, any differences in performance can be attributed to the adaptive nature of the controller (and not initial signal timings). All three traffic control parameters remain constant in FIXED. The fixed-time traffic signals have a cycle length of 80 seconds, and 87.5% of that is allotted to the *split*. The *split* is 35 seconds (per phase) and *offset* is zero. The complete two-phase traffic signal timing is shown in Table 5.3.

For the medium cycle, Webster's [125] optimal cycle length equation for minimising delay, Eq. (5.3), is used to determine the cycle length. The optimal cycle length is defined as:

$$C_{opt} = \frac{1.5t_L + 5}{1 - Y} \tag{5.3}$$

where the total lost time t_L in the cycle is defined as:

$$t_L = l_1 + l_2 \tag{5.4}$$

and Y is defined as:

$$Y = \frac{V_c}{s} \tag{5.5}$$

Cycle	
Phase 1	
Duration	*Light*
35	Green
4	Amber
1	Red
Phase 2	
Duration	*Light*
35	Green
4	Amber
1	Red

TABLE 5.3: Traffic signal timing for FIXED. This signal timing is also the initial traffic signal settings for the adaptive mechanisms (i.e., SCOOT, SUPRL, SAT/Q and GRACE)

where V_c is the flow rate (vph) of the critical lane and s is its saturation (vph). Lost time is due to vehicles starting up when the light first turns green and coming to a complete stop when the light turns red (clearance lost time). The start up lost time l_1 is normally set to 2 seconds [18]. While the clearance lost time l_2, from [18], is defined as

$$l_2 = y + ar - e \tag{5.6}$$

where y is the amber interval, ar is the *all red* interval and e is the estimated amount of amber and *all red* time that vehicles are in the intersection. The default value for e is 2 seconds [18]. For amber and red times, the minimum amber and red times suggested in [21] are used. Finally, all the remaining time in the cycle is given to the green interval.

Given the road geometry (and indirectly the driver agent parameters) and suggested default values from [21], the following values were used to calculate the fixed-time traffic signal *cycle* length[2]:

$$\begin{aligned} l_1 &= 2 \\ y &= 8 \\ ar &= 2 \\ e &= 2 \\ V_c &= 1080 \\ s &= 1417 \end{aligned}$$

[2]The final ideal cycle length value was rounded to 80 seconds.

5.8 Measures of Efficiency

The performance of all traffic controllers considered in this thesis is evaluated using three categories of metrics: *Travel Time*, *Traffic Density* and *Number of Stops. Travel time* is by far the most common way of measuring the effectiveness of traffic controllers and is computed for an individual vehicle as the amount of time it takes for the vehicle to complete its journey; lower travel times are better. *Traffic Density* is a measure of the amount of traffic, per kilometre, in a road network. *Number of Stops* is a measure of how the traffic is flowing and is computed as the number of vehicles that are not moving at any given time step; if more vehicles are moving, hence lower numbers of vehicles stopped, then the whole traffic system is operating more smoothly. Each metric is examined in several different forms.

First, the following *Travel Time* metrics:

- *Average Travel Time* (ATT) —average travel time of vehicles across the 30 simulations, which gives an overall measure of the effectiveness of the traffic control system;
- *Cumulative Average Travel Time* (CATT) —the cumulative average travel time as the simulation executes, which gives a measure of how the traffic control system performs over time particularly as traffic conditions change; and
- *Average Travel Time on Arrival* (ATTA) —the average travel time of the group of vehicles that have finished their trip at each time step, which gives a measure of how well the traffic control system responds to disruptions in the system, both with respect to the moment after the disruption as well as how well the system recovers after the disruption has passed.

Second, the following *Traffic Density* metrics:

- *Average Traffic Density* (ATD) —average traffic density across the 30 simulations, which gives an overall measure of how much traffic is trapped on the roadway. The average traffic density (over 30 simulations) is also examined on a single artery, however, this is expressed in terms of time, i.e., a ATD-time graph; and
- *Cumulative Average Density* (CAD) —the cumulative average traffic density as the simulation executes, which gives a measure of how traffic density in the system performs over time and as traffic conditions change. Traffic density is also examined on the north bound lane of the second artery.

Lastly, the following *Number of Stops* metrics:

- *Average Number of Stops* (ANS) across the 30 simulations, which gives an overall measure of how smoothly the traffic flows; and

- the *Cumulative Average Number of Stops* (CANS) —the cumulative average number of stops as the simulation executes, which gives a measure of how the system performs over time and as traffic conditions change.

5.9 Mann-Whitney Test

The Mann-Whitney test [126] is the non-parametric version of the independent t-test. It is used to test whether two samples come from the same population. In this thesis, statistical analysis of results were carried out using R [127] statistical programming language. The threshold value $\alpha = 0.05$ was used to determine whether the null hypothesis was rejected. The Mann-Whitney results are presented visually in a heat map, an example is shown in Figure 5.8. Figure 5.8 shows a series of pairwise comparisons of samples ABC, DEF, GHI, JKL. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance. The squares along the diagonal (upper left to lower right) are light because those represent a comparison of one sample to itself. In Figure 5.8, samples ABC and GHI significantly different while samples DEF and GHI are not.

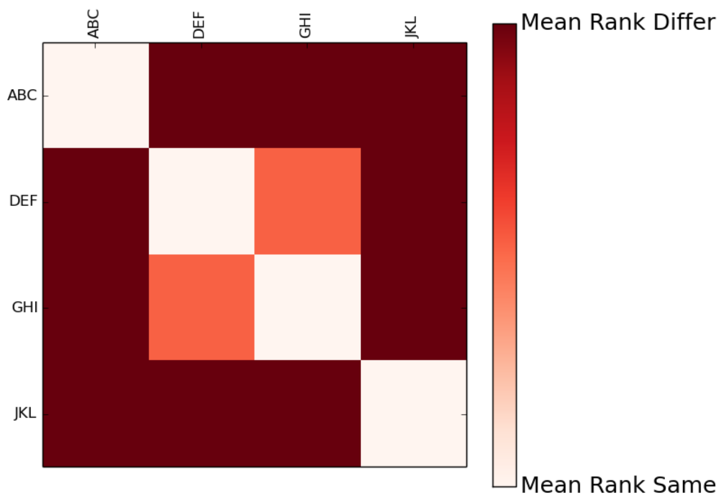


FIGURE 5.8: Sample pairwise Mann-Whitney test heat map results for samples ABC, DEF, GHI, JKL. The p-values from each test is represented as a coloured square, where dark squares denote statistical significance.

5.10 Summary

Table 5.4 list all of the traffic control systems presented in this chapter and the traffic control parameters that they adjust. A total of 252 experiments were executed using 21 mechanism, 6 traffic scenarios and 2 maps, Table 5.5. Each experimental conditions was repeated 30 times ("runs") to attain suitable statistics. Each simulation run lasted a

	Traffic Control Parameters		
Mechanism	*Split*	*Cycle*	*Offset*
SCOOT	*Event*	*Periodically*	*Event*
SCOOT(S)	*Event*	-	-
SCOOT(C)	-	*Periodically*	-
SCOOT(O)	-	-	*Event*
SCOOT(OC)	-	*Periodically*	*Event*
SCOOT(SC)	*Event*	*Periodically*	-
SCOOT(SO)	*Event*	-	*Event*
FIXED	-	-	-
SUPRL	*Periodically*	-	-

TABLE 5.4: Traffic control systems used as benchmarks as well as SCOOT variants. Traffic control parameters labelled *Periodically* updated periodically and thosed labelled *Event* means the time span in between adjustments fluctuates.

maximum of $15,000$ seconds (4 hours and 10 minutes); simulations could terminate early if all vehicles reached their destination before the maximum time had passed. Data, for the cumulative averages, is not collected until after the $1,000^{th}$ second. This allows traffic levels to reach a critical point so that averages more accurately reflect actual performance.

Mechanisms	*Traffic Scenarios*	*Maps*
DC2	Structured	Phoenix
FIXED	Unstructured	Portland
GRACE (7 variants)	Directional	
SUPRL	Regional	
SCOOT (+6 variants)	Football	
SAT	Constant	
SATQ		

TABLE 5.5: List of mechanisms, traffic flows and maps used in experiments. The variants are all possible combinations of using *split*, *offset* and *cycle* length for signal timing adjustments.

Chapter 6

Results of SAT/Q and MMDOS Experiments

6.1 Introduction

This chapter presents the traffic simulation results for SAT/Q and MMDOS. SAT and SATQ are the first set of market-based multi-agent traffic control system implemented to establish and test the framework that would be the bases for all the other market-based mechanisms. MMDOS is the second generation of my market-based traffic control system which is able to make finer adjustment to green time. Also, unlike SAT/Q which only adjust green time, MMDOS adjusts all three traffic control parameters. SAT, SATQ and MMDOS are compared with three other traffic control systems: FIXED, SCOOT & SUPRL. FIXED does not adjust any traffic control parameters, that is, the traffic signal timing is static. SCOOT is an adaptive traffic control system which adjusts all three traffic control parameters. Lastly, SUPRL is a Reinforcement-learning based traffic controller which only adjust the *split*. The mechanisms, traffic scenarios and maps presented in this chapter are shown in Table 6.1.

Mechanisms	*Traffic Scenarios*	*Maps*
SAT	Structured	Phoenix
SATQ	Unstructured	Portland
MMDOS	Directional	
FIXED	Regional	
SCOOT	Football	
SUPRL	Constant	

TABLE 6.1: List of mechanisms, traffic flows and maps presented in this chapter.

This chapter is organised into two major parts, the first presents and analyses the results for each map, Phoenix (Section 6.2) and Portland (Section 6.3), and the second, is analysis across both maps (Section 6.4). Traffic performance is measured using three metrics: *average travel time*, *traffic density*, and *number of stops*. Thus, Section 6.2

presents results from simulations executed on the Phoenix map and is divided into three sub-sections, one for each metric. Likewise, Section 6.3 is divided into three sub-sections, one for each metric, but for simulations executed on the Portland map. Furthermore, traffic performance is evaluated using six traffic scenarios: *structured*, *unstructured*, *football*, *directional*, *constant*, and *regional*.

Lastly, Section 6.4 contains a summary of the results and addresses the following hypothesis:

Hypothesis 1 *There will be a significant difference in* ATT *of SAT compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

Hypothesis 2 *There will be a significant difference in* ATD *of SAT compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

Hypothesis 3 *There will be a significant difference in* ANS *of SAT compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

Hypothesis 4 *There will be a significant difference in* ATT *of SATQ compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

Hypothesis 5 *There will be a significant difference in* ATD *of SATQ compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

Hypothesis 6 *There will be a significant difference in* ANS *of SATQ compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

Hypothesis 7 *There will be a significant difference in* ATT *of MMDOS compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

Hypothesis 8 *There will be a significant difference in* ATD *of MMDOS compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

Hypothesis 9 *There will be a significant difference in* ANS *of MMDOS compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

Average Travel Time (ATT) (*std.*)

Mechanism	*Structured*	*Regional*	*Constant*
FIXED	166.11 (1.09)	184.07 (1.32)	184.6 (0.2)
SAT	375.7 (9.73)	228.62 (5.32)	216.72 (3.81)
SATQ	183.51 (3.58)	190.55 (3.12)	209.66 (4.47)
MMDOS	150.27 (2.72)	144.28 (0.51)	190.83 (11.86)
SCOOT	**144.8 (3.44)**	**129.42 (3.71)**	**144.7 (3.52)**
SUPRL	159.48 (1.3)	144.03 (1.42)	206.12 (7.85)
	Unstructured	*Football*	*Directional*
FIXED	1108.81 (168.99)	190.89 (12.8)	173.18 (0.98)
SAT	1748.95 (723.67)	537.44 (11.08)	360.66 (9.23)
SATQ	**604.81 (20.65)**	174.5 (6.74)	193.87 (4.75)
MMDOS	717.92 (79.4)	154.28 (4.87)	160.17 (3.49)
SCOOT	1231.36 (369.63)	184.81 (7.66)	**146.93 (5.16)**
SUPRL	855.66 (78.43)	**142.76 (4.05)**	160.4 (1.26)

Table 6.2: Average travel times (ATT) for each mechanism and traffic scenario.

6.2 Results: Phoenix

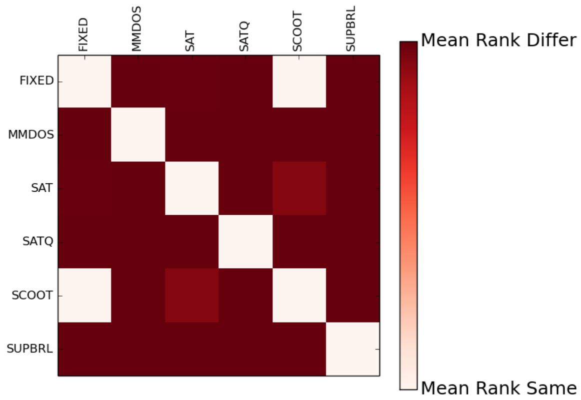
This section presents the results of the experiments executed on the Phoenix map. This section is divided into sub-sections, covering each of the three traffic performance metrics: *average travel time* (Section 6.2.1), *traffic density* (Section 6.2.4), and *number of stops* (Section 6.2.7). The traffic control systems are evaluated in three traffic scenarios with *predictable* traffic flow (*structured*, *regional*, and *constant*) and three traffic scenarios with *unpredictable* traffic flow (*unstructured*, *football*, and *directional*). Results for ATTA (Section 6.2.3) and traffic density on a major artery (Section 6.2.6) are presented for SATQ, MMDOS, SCOOT and FIXED in the *unpredictable* traffic scenarios. SAT is not included in Sections 6.2.3 and 6.2.6 because of its poor ATT performance. Also, Sections 6.2.3 and 6.2.6 only cover the traffic scenarios with *unpredictable* traffic flow because they provide better conditions for illustrating the differences between the traffic control mechanisms. The Mann-Whitney test is used to determine statistical significance between traffic performance results. The threshold value of $p = .05$ was used to determine whether the null hypotheses (the samples were the same) was rejected. The Mann-Whitney test results are presented in a visual manner in lieu of tables to provide the same information but in a more compact manner than a large table(s).

6.2.1 Travel Time (ATT)

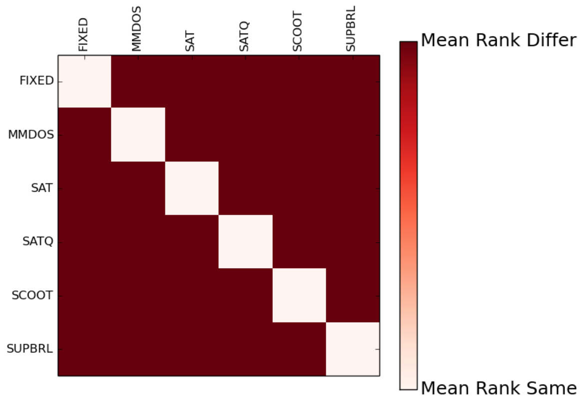
In the *unstructured* traffic scenario, SATQ has the lowest average travel times, Table 6.2. The difference in ATT performance between SATQ and the other mechanisms is significant, see Figure 6.1a. MMDOS has the second lowest ATT in the *unstructured* traffic scenario and results are significant as well, see Figure 6.1a. Although not statistically significant, SCOOT has higher ATT than FIXED in *unstructured* traffic.

In the *football* scenario, MMDOS and SATQ have a lower ATT than SCOOT and FIXED but not SUPRL. Figure 6.1c shows that in the *football* scenario, the difference in performance is significant. SUPRL has the lowest ATT in the *football* scenario. In the *directional* traffic scenario, SCOOT has the lowest ATT. Figure 6.1e shows that MMDOS and SUPRL have statistically similar ATT in *directional* traffic. Also, in the *directional* traffic scenario, SATQ has higher ATT than SCOOT, SUPRL and FIXED.

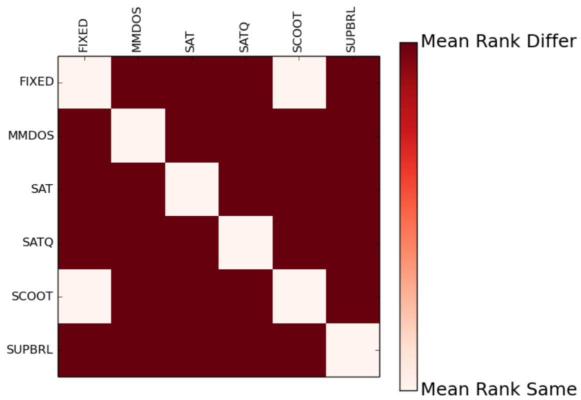
Tables 6.2 also shows that SCOOT has the lowest ATT in all three traffic scenarios with *predictable* traffic. Also, SATQ has a higher ATT than the benchmarks in *structured*, *regional* and *constant* traffic. MMDOS performs as well as or better than SUPRL in each of the traffic scenarios with *predictable* traffic. For example, in *structured* traffic, MMDOS has lower ATT than SUPRL and in *regional* MMDOS perform as well as SUPRL, see Figure 6.1d. Lastly, SAT has the highest ATT in all six scenarios on the Phoenix map.



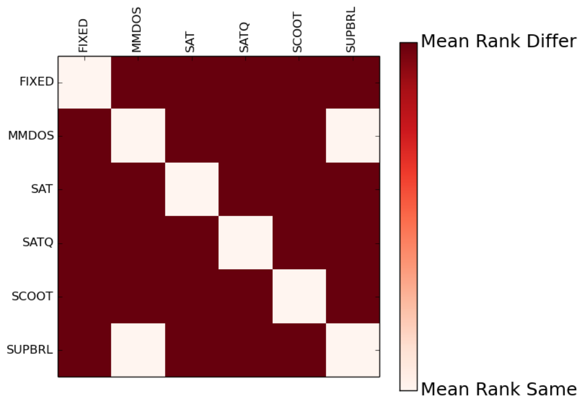
(a) Unstructured.



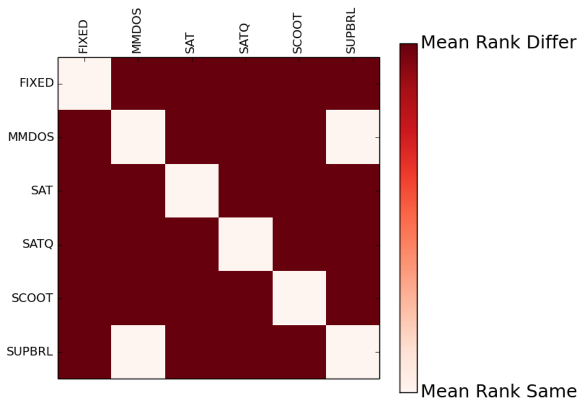
(b) Structured.



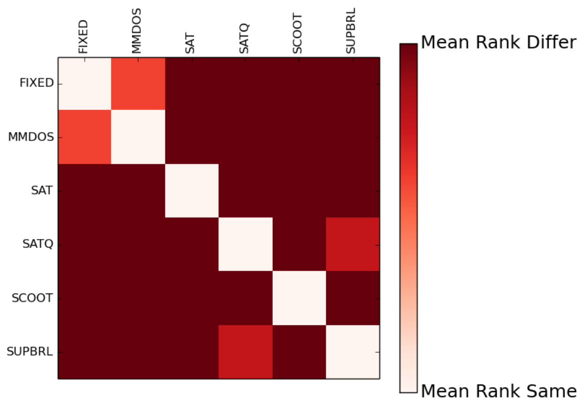
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 6.1: Visual representation of two-sample Mann-Whitney test conducted on ATT (Phoenix map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

6.2.2 Cumulative Average Travel Time (CATT)

This section examines the *cumulative average travel time* (CATT) during each scenario, Figure 6.2a, shows how well SATQ performs on the Phoenix map in *unstructured* traffic. Although all the mechanisms experienced an increase in travel times during the disruption, SATQ maintains the lowest trip times from the very beginning of the simulation. The other mechanisms have similar travel time before and during the disruption, SAT is the exception, it has a sharp rise in travel times and continues to increase until the simulation terminates. Figure 6.2a shows that initially MMDOS performs as well as the benchmarks, however, once the disruption has terminated MMDOS has lower CATT than SCOOT, FIXED and SUPRL. Also, SUPRL departs from the trend exhibited by SCOOT and FIXED shortly after the disruption ends.

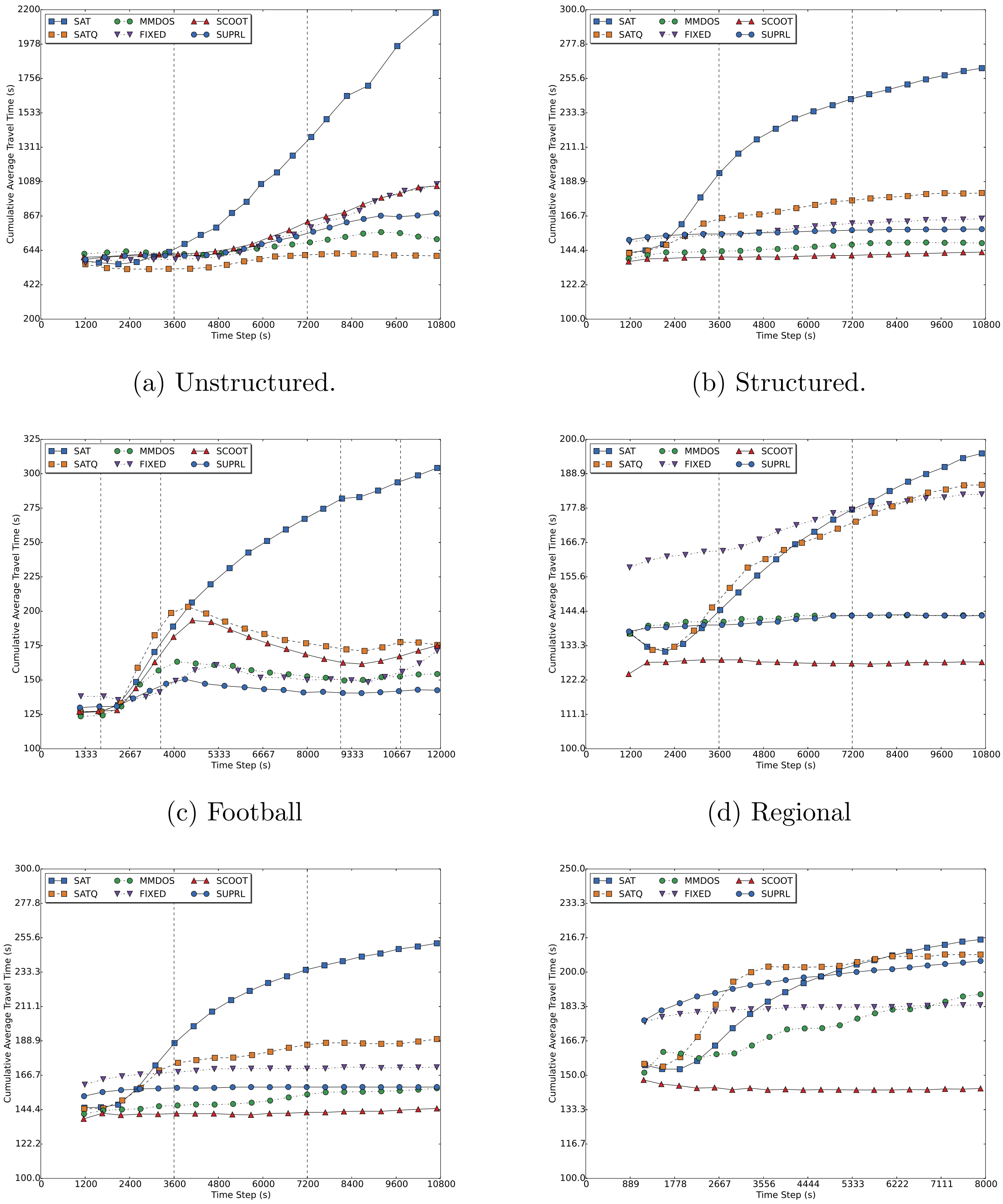
In the *football* traffic scenario, the mechanisms display distinct behaviours in terms of CATT (see Figure 6.2c). Figure 6.2c shows that SATQ's increase in CATT during the first disruption is similar to the increase in CATT of SCOOT and SAT. The CATT of MMDOS, on the other hand, does not increase as much as SAT/Q in the first disruption. SCOOT and SATQ are able to recover, somewhat, during the match but SAT's travel times continue to rise for the duration of the simulation. Figure 6.2c also shows that FIXED and SUPRL have only slight increases in travel times during the first disruption (influx of traffic) and then remain fairly the same until the second disruption where travel times increase further for FIXED. Also, in the second disruption, MMDOS displays little change in CATT.

In *directional* traffic, Figure 6.2e, all the mechanisms, except SAT, quickly reach their maximum levels of CATT and then plateau, displaying little or no change during the disruption. SAT is the sole mechanism to display high levels (above 200 seconds) of CATT in *directional* traffic. In *directional* traffic, SCOOT has the lowest CATT throughout the scenario. However, initially MMDOS has CATT close to SCOOT levels but, after the disruption, CATT of MMDOS reaches the same level as SUPRL.

In the *structured* traffic scenario, Figure 6.2b, FIXED and SUPRL have similar CATT. Although not as low as SCOOT's CATT, in *structured* traffic, MMDOS has CATT lower than FIXED, SUPRL and SATQ. In *regional* traffic, Figure 6.2d, the peak travel time for SCOOT occurs early on in the simulation and does not rise during the disruption. SCOOT maintains low trip times throughout the entire simulation. Although SUPRL and MMDOS have slightly higher CATT than SCOOT, both mechanisms behave in a similar manner to SCOOT in *regional* traffic. In *regional* traffic, the CATT of SUPRL and MMDOS plateaus early on and shows little change even during the disruption (see Figure 6.2d). Lastly, in the *regional* traffic scenario SAT & SATQ display nearly identical growths of travel times.

On the Phoenix map SCOOT has the lowest CATT in the *constant* traffic scenario. In the *constant* traffic scenario, SAT and SATQ have a sharp increase in CATT prior to the disruption (see Figure 6.2f). Initially, in *constant* traffic, SATQ has higher CATT than SAT, however, during the disruption the CATT of SAT surpasses SATQ and SUPRL.

Although the CATT of MMDOS increases during the disruption, it remains lower than SAT, SATQ and SUPRL (see Figure 6.2f).



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

(e) Directional

(f) Constant

FIGURE 6.2: Cumulative average travel times (over 30 simulations) on Phoenix map. Beginning and ending of disruptions are marked by dotted lines.

6.2.3 Average Travel Time on Arrival (ATTA)

An examination of *average travel time* at each time step reveal how much better SATQ performs in *unstructured* traffic than the other mechanisms. Figure 6.3 contains the ATTA for SATQ, SCOOT and SUPRL. Vehicles under SATQ control have *average travel times* under 800 seconds throughout the entire simulation, even during the one hour disruption. SCOOT, during the disruption in *unstructured* traffic, has an increase in trip times that continues on after the disruption. However, a look at ATTA with SUPRL reveals it too experiences an increase in travel time midway through the disruption but not as elevated as SCOOT. In *unstructured* traffic, the ATTA of MMDOS is similar to SUPRL and SCOOT prior to the disruption (see Figure 6.4). During the *unstructured* traffic disruption, the ATTA of MMDOS increases with a small group of vehicles maintaining lower ATTA than SCOOT and SUPRL. The majority of vehicles that completed their journey after the *unstructured* disruption have lower ATTA than SCOOT and SUPRL.

Although SATQ has a lower over all average travel time than SCOOT in the *football* scenario, Figure 6.5 shows that there are many instances where vehicles under SATQ control experienced travel times far greater than SCOOT. During the first disruption and the *football* match many SATQ vehicles have higher travel time than SCOOT and SUPRL. Figure 6.5 also shows that during the football match and second disruption SATQ has groups of vehicles with lower ATTA than SUPRL. In the *football* scenario, the ATTA of MMDOS resembles the ATTA of SUPRL, especially during the disruptions. Also, the ATTA of MMDOS during the football match is within the upper bound of SUPRL's ATTA.

In *directional* traffic with SATQ in control many vehicles have low ATTA (sub 100 seconds), however, some vehicles have much higher ATTA (above 250 seconds), see Figure 6.7. The ATTA of SUPRL form two narrow clusters both of which are 250 seconds. In *directiona* traffic, the lower bound of the ATTA of MMDOS increases but not the upper bound, i.e., the highest ATTA in *directional* traffic with MMDOS, remains fairly the same. SCOOT's ATTA in *directional* traffic is better on the Phoenix map. Figure 6.7 shows that during the disruption, the range of ATTA of SCOOT becomes more narrow and lessens.

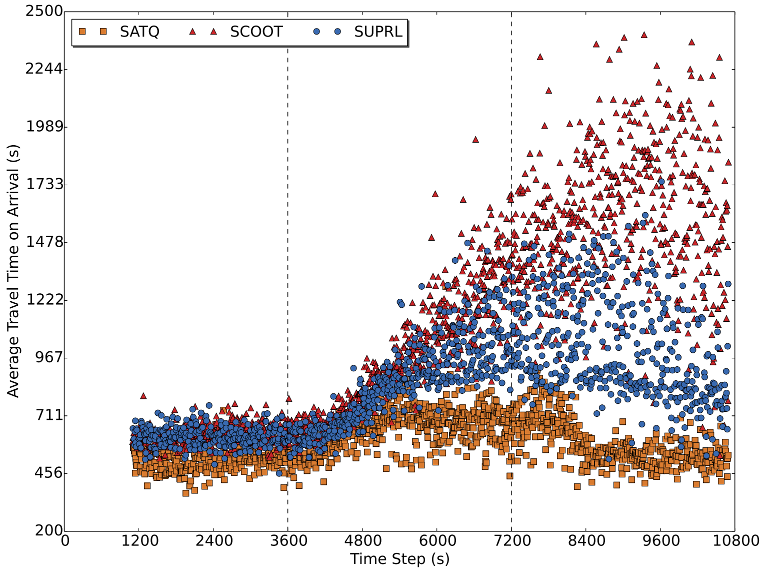


FIGURE 6.3: The graph shows the average travel times of vehicles that have completed their journey at each time step. (over 30 simulations) in *unstructured* traffic on the Phoenix map.

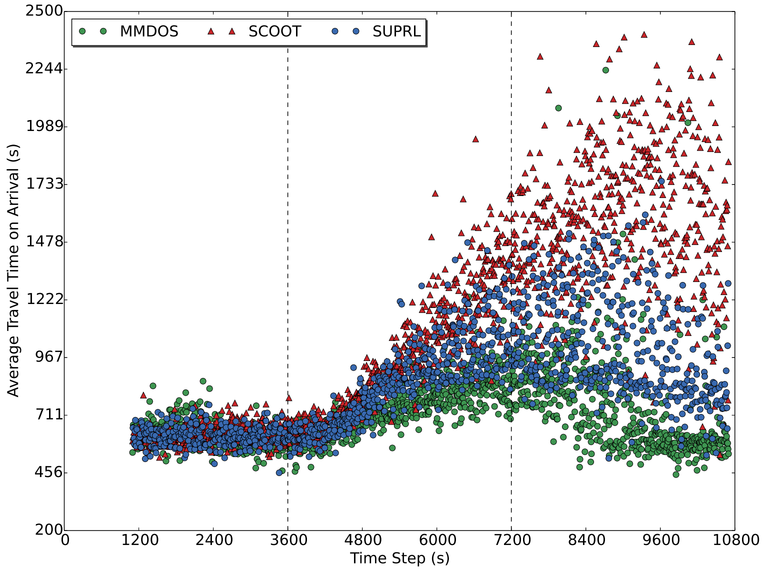


FIGURE 6.4: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *unstructured* traffic on the Phoenix map.

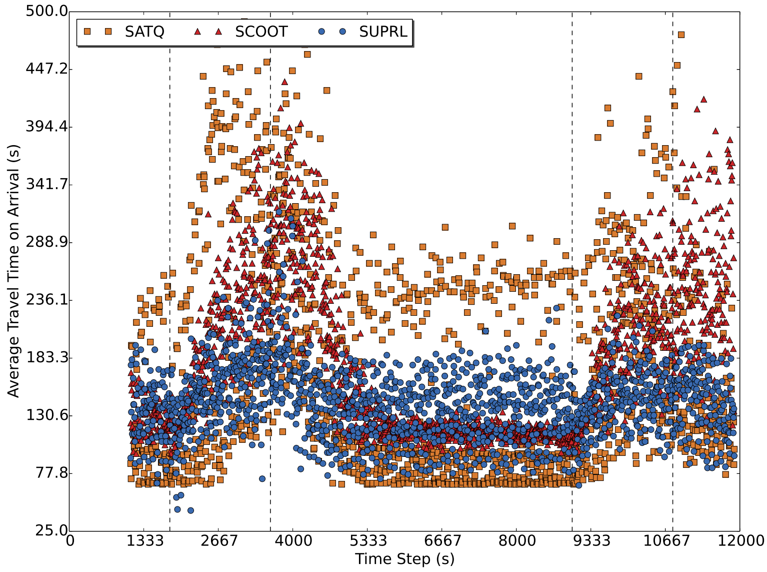


FIGURE 6.5: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *football* traffic on the Phoenix map.

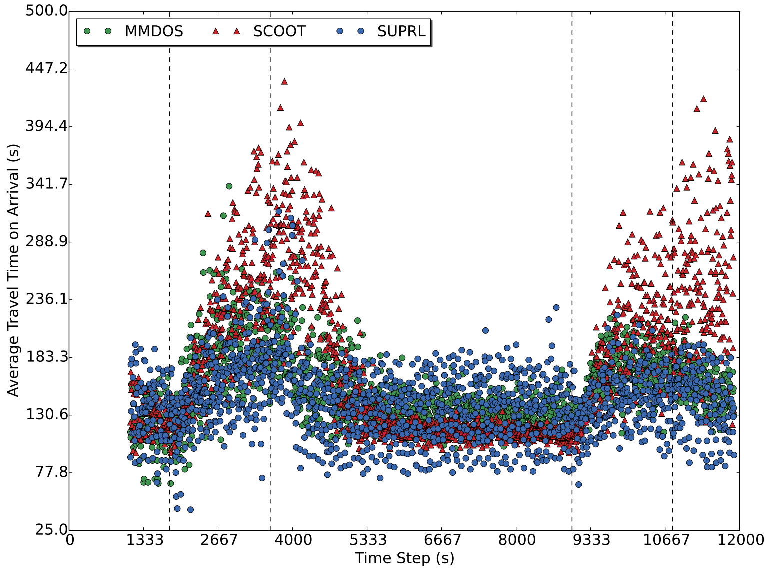


FIGURE 6.6: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *football* traffic on the Phoenix map.

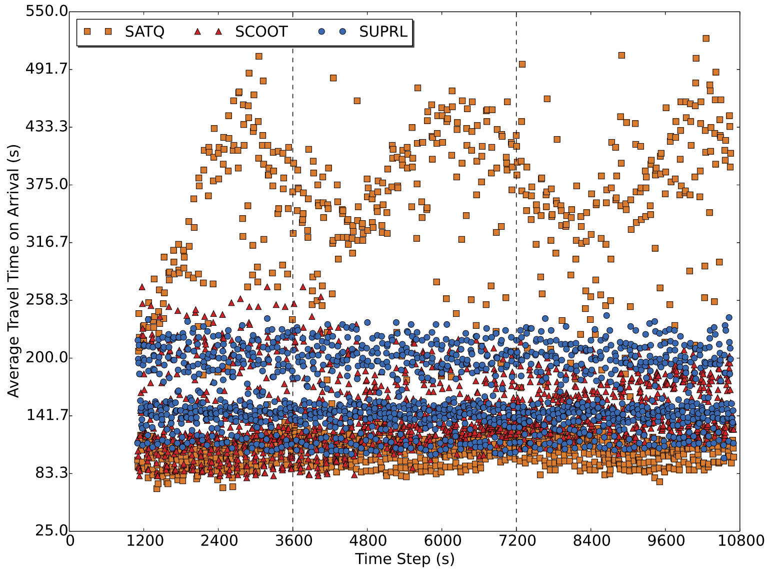


Figure 6.7: The graph shows the average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *directional* traffic on the Phoenix map.

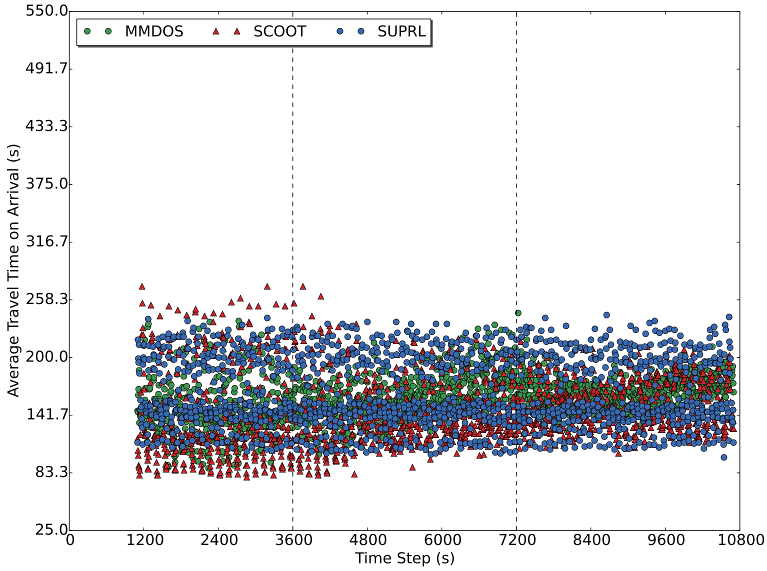


Figure 6.8: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *directional* traffic on the Phoenix map.

6.2.4 Density (ATD)

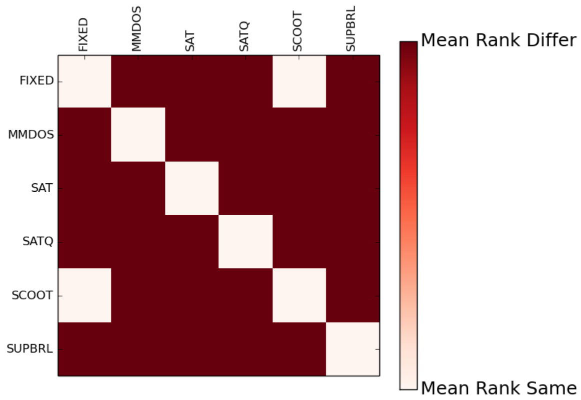
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	12.39 (0.19)	16.22 (0.25)	21.44 (0.09)
SAT	16.34 (0.5)	15.88 (0.55)	23.29 (0.49)
SATQ	13.76 (0.35)	16.61 (0.33)	26.2 (0.56)
MMDOS	11.49 (0.24)	13.05 (0.14)	23.6 (1.27)
SCOOT	**11.07 (0.26)**	**11.72 (0.38)**	**18.07 (0.44)**
SUPRL	12.15 (0.17)	13.1 (0.17)	25.51 (1.01)
	Unstructured	*Football*	*Directional*
FIXED	10.46 (1.31)	8.08 (0.51)	13.5 (0.15)
SAT	32.28 (3.12)	15.42 (0.38)	16.5 (0.46)
SATQ	**6.2 (0.33)**	7.51 (0.32)	15.24 (0.4)
MMDOS	7.82 (3.33)	6.64 (0.24)	12.72 (0.35)
SCOOT	11.8 (3.42)	7.51 (0.33)	**11.75 (0.39)**
SUPRL	8.53 (0.86)	**6.13 (0.21)**	12.85 (0.2)

Average Traffic Density (ATD) (*std.*) — **Traffic Pattern**

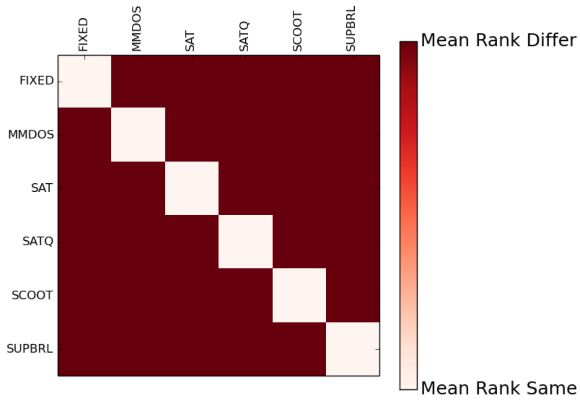
TABLE 6.3: Average traffic density (ATD) for each mechanism and traffic scenario.

Table 6.3 contains the ATD results for each mechanism and traffic scenario. In *unstructured* traffic, SATQ has the lowest ATD the results are significant (see Figure 6.9a). Additionally, MMDOS has the second lowest ATD in *unstructured* traffic which is also significant. However, in the *football* scenario, SATQ performs as well as SCOOT but not better than SUPRL. On the Phoenix map SCOOT has the lowest ATD in *directional* traffic, MMDOS also has the second lowest ATD in the *football* and *directional* traffic scenario. Figures 6.9c and 6.9e show that the difference in ATD of the market-based traffic control systems and the benchmarks is significant in the *football* and *directional* traffic scenario.

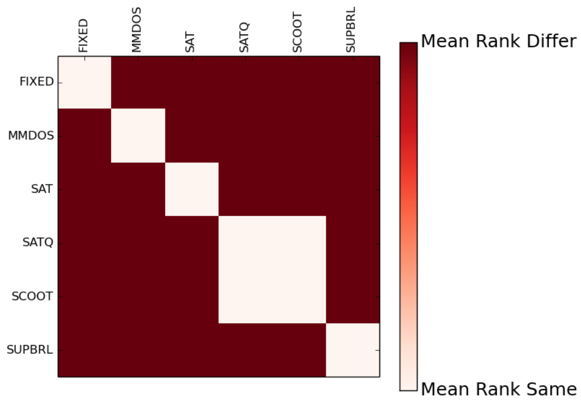
The remainder of the scenarios reflect the ATT results in that SCOOT has the lowest ATD, Figure 6.9 shows that in each scenario, SCOOT's performance is significantly different from the other mechanisms. In the *structured* traffic scenario, MMDOS has the second highest ATD followed by SATQ. In *regional* traffic, MMDOS and SUPRL have statistical similar ATD (see Figure 6.9d). Although SAT has the highest ATT in *regional* and *constant* traffic, it has lower ATD than SATQ in both traffic scenarios. Lastly, in *constant* traffic on the Phoenix map, SAT and MMDOS have lower ATD than SATQ and SUPRL.



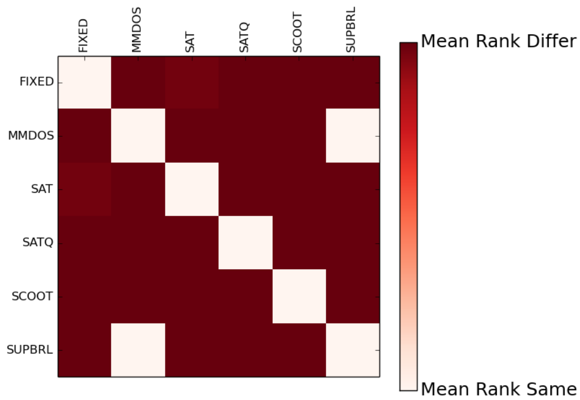
(a) Unstructured.



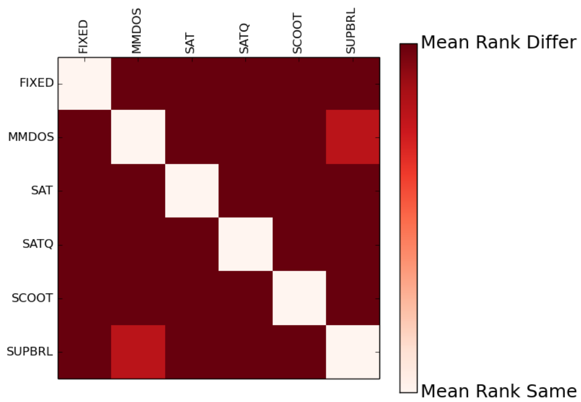
(b) Structured.



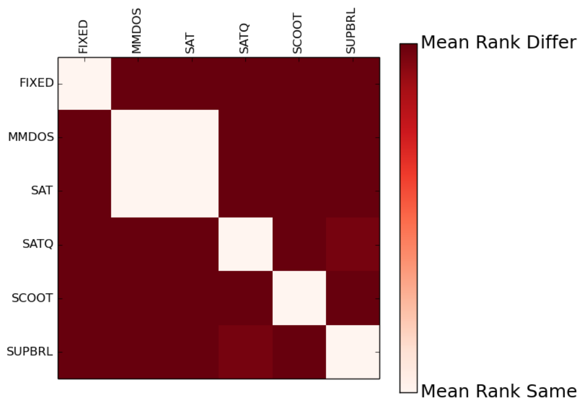
(c) Football



(d) Regional



(e) Directional



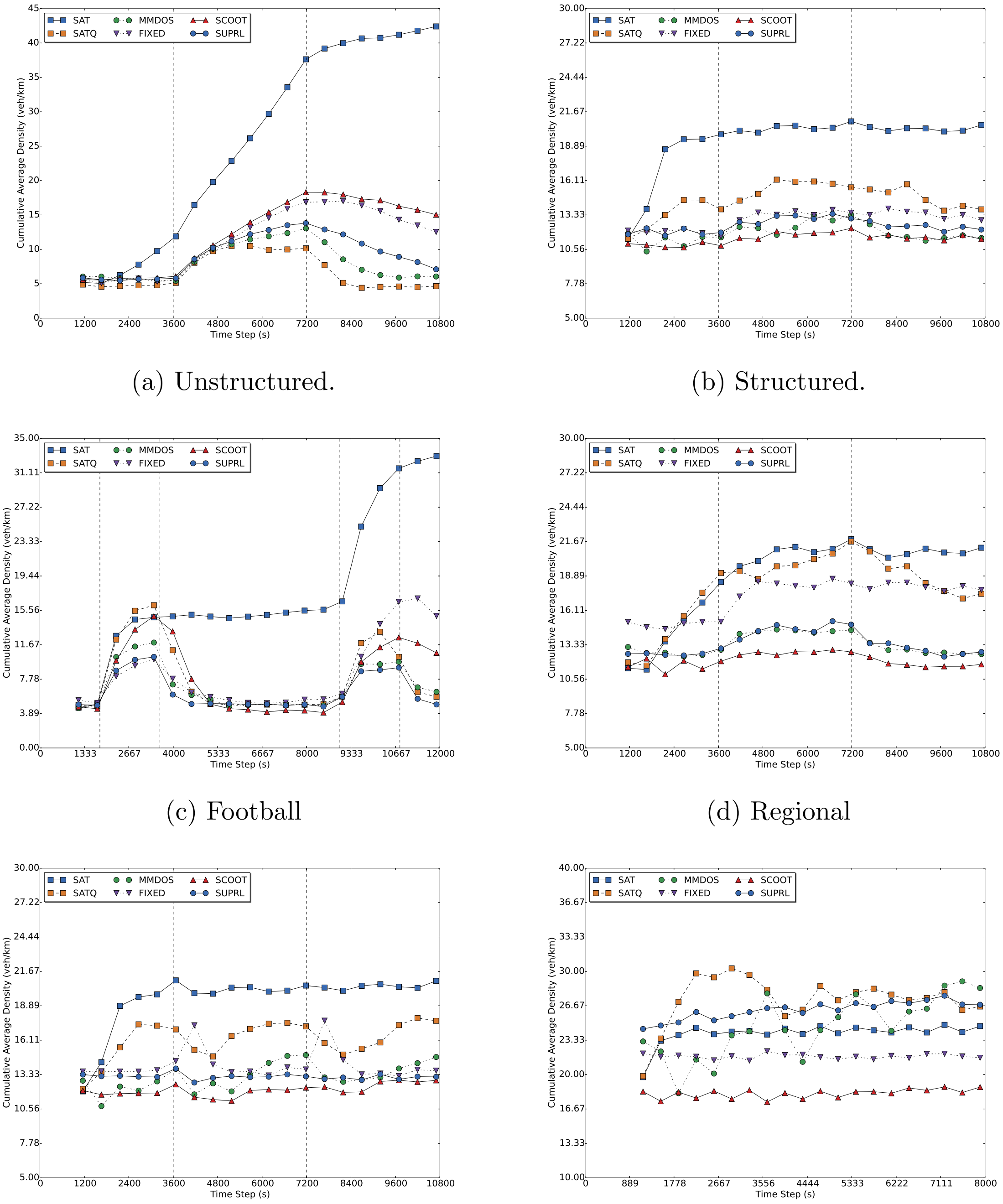
(f) Constant

FIGURE 6.9: Visual representation of two-sample Mann-Whitney test conducted on ATD (Phoenix map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

6.2.5 Cumulative Average Density (CAD)

In *unstructured* traffic, Figure 6.10a, shows that all the mechanisms except for SAT have very similar CAD profiles until the middle of the disruption. Half way through the disruption the mechanisms begin to show distinct levels of traffic density, listed from least to greatest: SATQ, SUPRL, FIXED, SCOOT and finally SAT. During the disruption, MMDOS and SATQ have a lower increase in CAD than SCOOT, FIXED and SUPRL. In addition, after the disruption, the decrease in CAD is greater with MMDOS and SATQ than the other mechanisms. SAT has the highest spike in CAD compared to all the other mechanisms and ended with traffic density level over twice that of SCOOT which has the second highest density levels (see Figure 6.10a). Figure 6.10b shows that in the *football* scenario, the mechanisms manage the disruptions differently. For example, during the first disruption, SAT, SATQ, and SCOOT have higher levels of traffic density than SUPRL, MMDOS and FIXED. Additionally, SUPRL, MMDOS and FIXED recover quicker from the first disruption. However, during the match the mechanisms, excluding SAT, display the same low, pre-disruption, CAD. Again, during the second *football* disruption, the reaction to increased traffic intensity is different from the first disruption. In the second disruption, MMDOS and SUPRL have lower CAD than the other mechanisms but FIXED displays a greater increase in CAD in comparison to its performance in the first disruption. Lastly, in the *football* scenario, after the first disruption SAT maintains a high level of traffic density with no recovery periods. In *directional* traffic, Figures 6.10e shows little change in CAD even during the disruption. SCOOT has the lowest CAD in *directional* traffic. MMDOS displays similar CAD to SUPRL in *directional* traffic. In all three scenarios with *unpredictable* traffic, SAT has the highest levels of CAD.

In *structured*, *regional* and *constant* traffic, SCOOT has the lowest CAD (see Figure 6.10). In *structured* traffic, MMDOS, FIXED and SUPRL have similar performance in terms of CAD. Additionally, there is little change in CAD during the disruption phase of the *structured* traffic scenario. However, in *regional* traffic, MMDOS and SUPRL have nearly identical CAD. Lastly, in *constant* traffic, Figure 6.10f, the mechanisms display more distinct CAD than in the other traffic scenarios. In *constant* traffic, excluding SATQ and MMDOS, all the mechanisms reach different peaks and display little change in CAD as the scenario progresses.



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

(e) Directional

(f) Constant

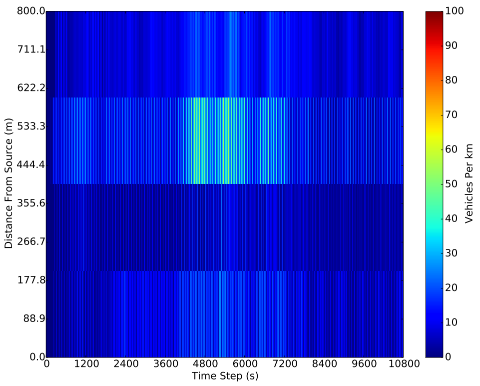
FIGURE 6.10: Cumulative average density (over 30 simulations) on the Phoenix map. Beginning and ending of disruptions are marked by dotted lines.

6.2.6 Density on Major Artery

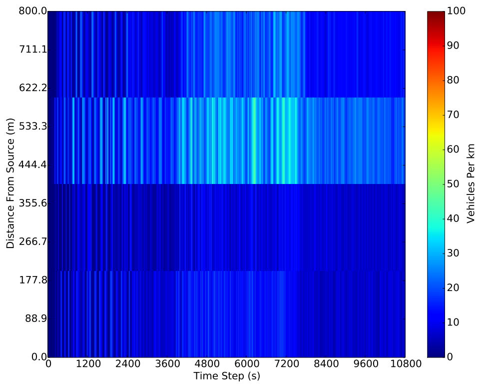
A closer look at traffic density on the second major artery reveals differences in mechanism performance and the effects of geographic properties of the road networks. The second major artery under SATQ's control with *unstructured* traffic, Figure 8.8a, has much less traffic than SCOOT does on the same artery shown in Figure 8.8d. However, SUPRL has the lowest levels of traffic density on the second artery (see Figure 8.8c). The majority of traffic on the second artery with MMDOS is on the third road segment, Figure 8.8b, which is similar to traffic density on the second artery for SUPRL (Figure 8.8c) and FIXED (see Figure 8.8f).

On the Phoenix map, MMDOS, SUPRL, and SATQ have similar traffic density on the second artery during the *football* scenario (see Figure 8.9). All three mechanisms display an increase in traffic density on the third road segment near the end of the scenario which dissipates before the scenario ends. In comparison, traffic density on the same artery is far higher with SAT, SCOOT and FIXED. However, unlike SCOOT and FIXED which have high traffic density which covers three fourth of the artery, SAT has vehicle trapped on two road segments only.

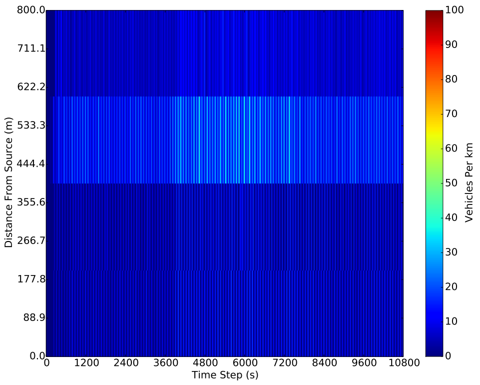
Figure 8.10 shows traffic density on the second artery in *directional* traffic for SAT/Q, MMDOS, SCOOT and SUPRL. None of the mechanisms show vehicles trapped on the artery during the disruption. That is, when the direction of the heavy flow reversed signal timing changes did not cause excessive queues to form. However, before and after the disruption, MMDOS, SUPRL and FIXED have higher traffic density downstream than SAT/Q and SCOOT. Additionally, all of the mechanisms display some *gating* behaviour, especially FIXED. In other words, heavy traffic density forms on the very first road segment while further downstream has less traffic.



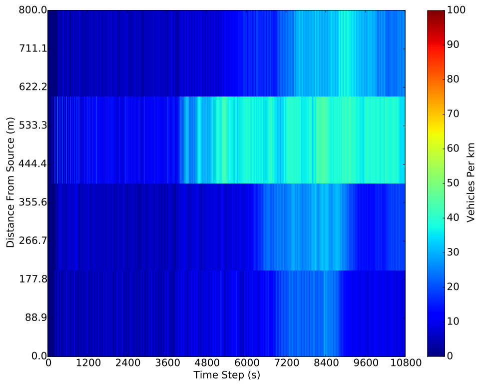
(a) SATQ



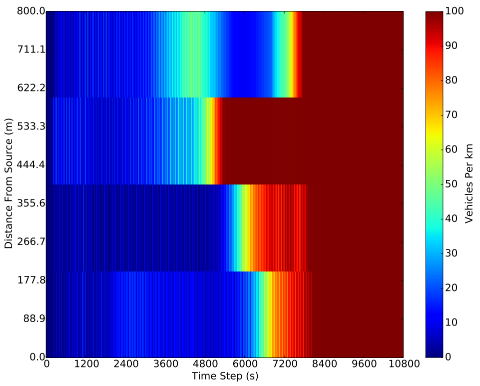
(b) MMDOS.



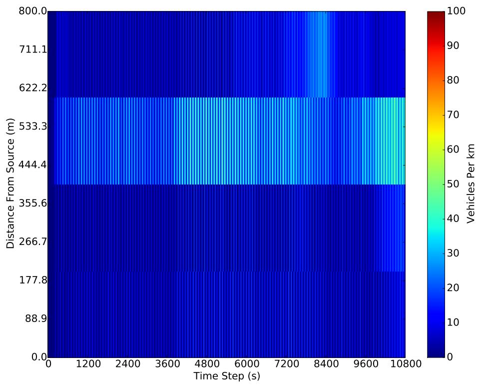
(c) SUPRL



(d) SCOOT

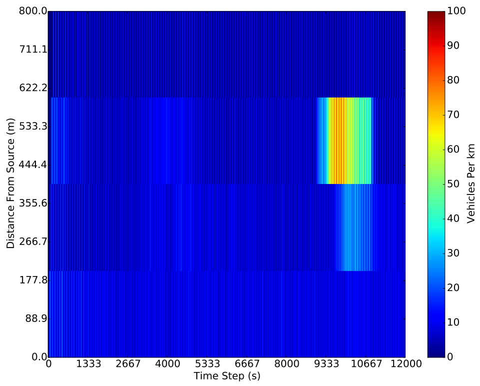


(e) SAT

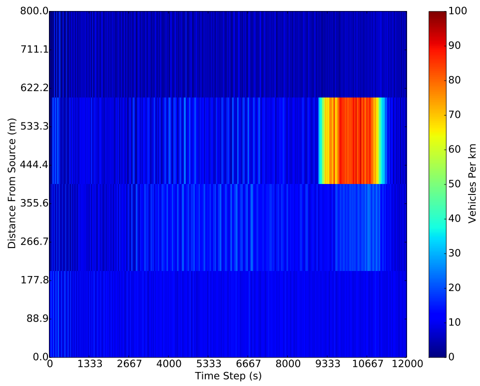


(f) FIXED

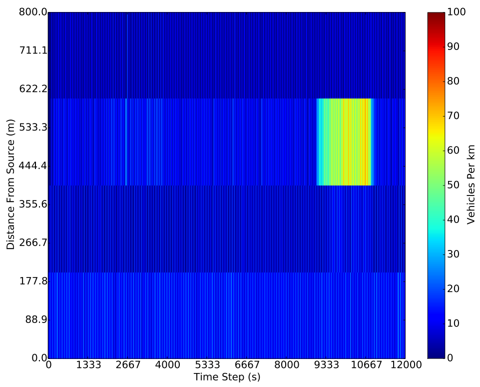
FIGURE 6.11: Traffic density on major artery in *unstructured* on Phoenix map.



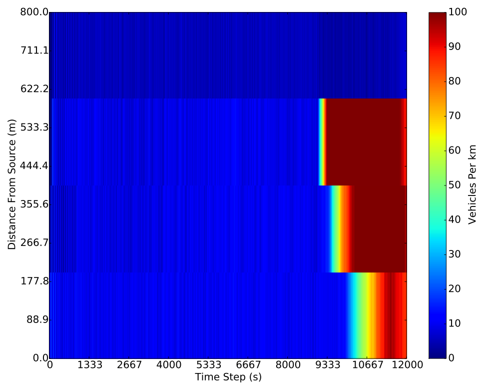
(a) SATQ



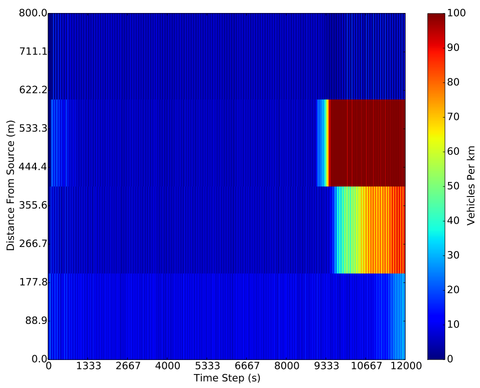
(b) MMDOS.



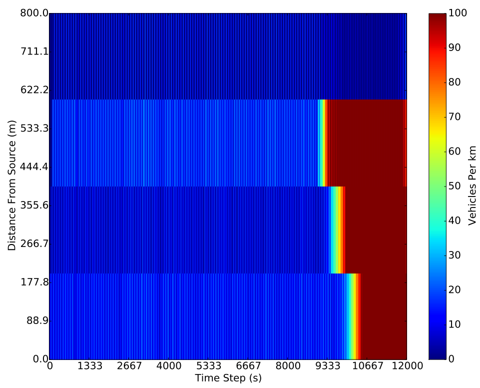
(c) SUPRL



(d) SCOOT

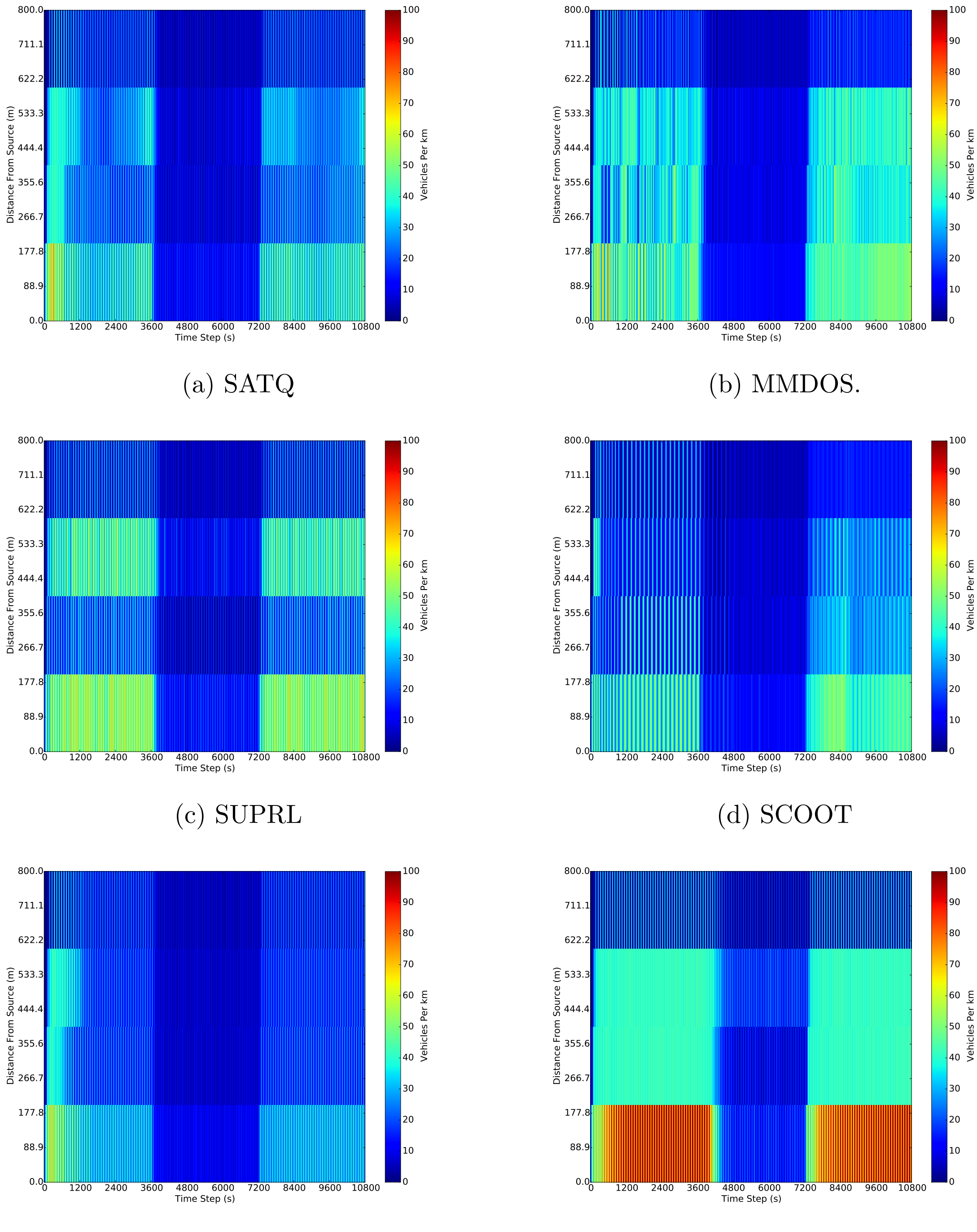


(e) SAT



(f) FIXED

FIGURE 6.12: Traffic density on major artery in *football* on Phoenix map.



(a) SATQ

(b) MMDOS.

(c) SUPRL

(d) SCOOT

(e) SAT

(f) FIXED

FIGURE 6.13: Traffic density on major artery in *directional* on Phoenix map.

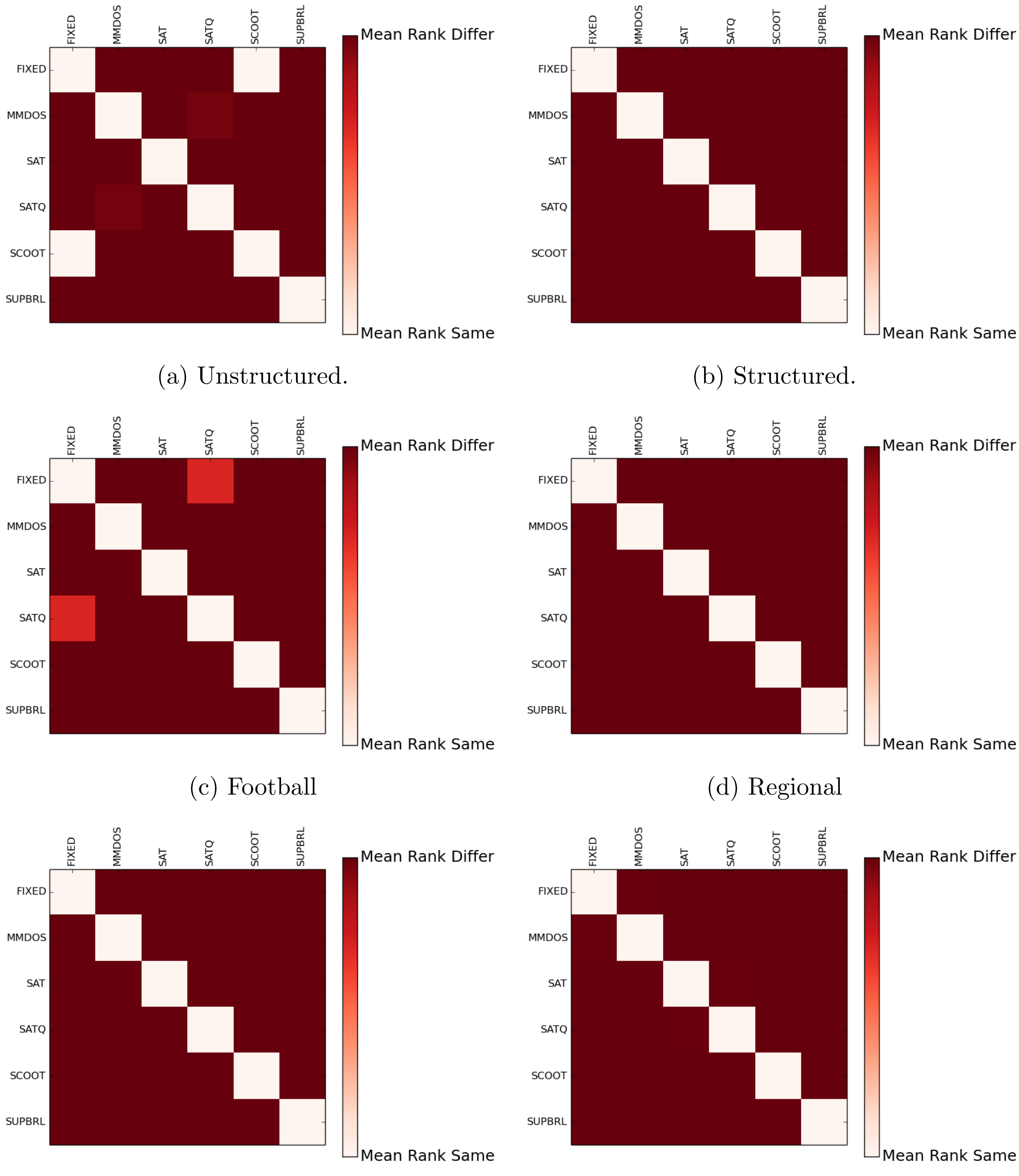
Average Number of Stops (ANS) (*std.*)			
		Traffic Pattern	
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	70.23 (1.33)	98.4 (1.92)	130.7 (0.6)
SAT	153.03 (5.58)	123.83 (5.33)	179.9 (5.55)
SATQ	87.77 (3.29)	113.57 (3.08)	174.47 (5.1)
MMDOS	**53.87 (1.57)**	**49.97 (0.76)**	138.57 (11.27)
SCOOT	60.17 (2.18)	53.97 (2.92)	**98.63 (3.6)**
SUPRL	66.63 (1.4)	63.03 (1.33)	160.23 (10.39)
	Unstructured	*Football*	*Directional*
FIXED	68.33 (16.6)	50.27 (5.01)	78.97 (1.13)
SAT	418.2 (41.34)	165.03 (4.87)	152.63 (5.02)
SATQ	**30.83 (2.61)**	47.2 (3.35)	100.83 (3.84)
MMDOS	44.57 (51.5)	33.9 (2.23)	**61.5 (2.49)**
SCOOT	89.97 (52.73)	38.5 (3.14)	64.57 (3.43)
SUPRL	46.6 (7.83)	**30.13 (1.81)**	68.87 (1.55)

TABLE 6.4: Average *number of stops* (ANS) for each mechanism and traffic scenario.

6.2.7 Vehicle Stops (ANS)

Table 6.4 contains the ANS for each mechanism in the six traffic scenarios completed on the Phoenix map. In *unstructured* traffic, SATQ has the lowest ANS and MMDOS the second lowest ANS; Figure 6.14a shows that both results are significantly different from the benchmarks. The *unstructured* traffic scenario is the only scenario where SATQ has the lowest ANS. SUPRL and MMDOS have the lowest ANS in the *football* and *directional* traffic scenarios, respectively, both results are significant, Figures 6.14c and 6.14e.

In two of the traffic scenarios with *predictable* traffic flows, (*structured* and *constant*), SCOOT has the lowest ANS. In the third scenario, *regional* traffic, MMDOS has the lowest ANS. Figure 6.14 shows that the difference in performance amongst the mechanisms in *predictable* traffic is significant. Additionally, in *constant* traffic, SATQ and SUPRL have higher ANS than FIXED. Lastly, SAT has significantly higher ANS than the other mechanisms in every traffic scenario.



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

(e) Directional

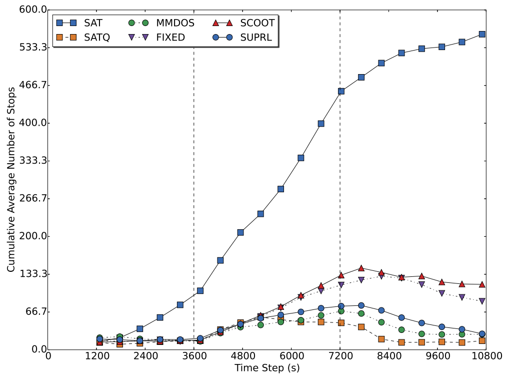
(f) Constant

FIGURE 6.14: Visual representation of two-sample Mann-Whitney test conducted on ANS (Phoenix map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

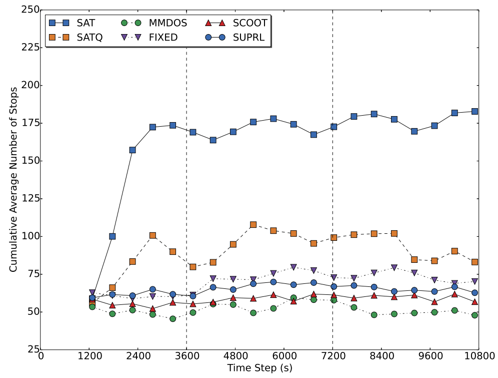
6.2.8 Cumulative Average Number of Stops (CANS)

Excluding SAT, all of the mechanisms have similar CANS during the initial phase of the *unstructured* traffic scenario (see Figure 6.15a). In *unstructured* traffic, on the Phoenix map, the CANS of SATQ and MMDOS does not increase as much as FIXED, SCOOT, SUPRL and SAT during the disruption. Figure 6.15a shows that with SATQ and MMDOS, CANS plateaus during the disruption and begin to recover afterwards (return to pre-disruption levels of CANS). In the *football* scenario, Figure 6.15c, during the football match, all the mechanisms (excluding SATQ) quickly reach their pre-disruption CANS. However, during the *football* traffic disruptions, the mechanisms display an increase in CANS. In the first disruption, SUPRL and FIXED have the lowest CANS, while in the second disruption MMDOS and SUPRL have the lowest CANS. In *directional* traffic, Figure 6.15e, MMDOS and SCOOT have the lowest CANS. During the *directional* traffic disruption MMDOS, SUPRL and SCOOT show little change in CANS. Lastly, on the Phoenix map, SAT and SATQ have greater CANS than all the other mechanisms in *directional* traffic.

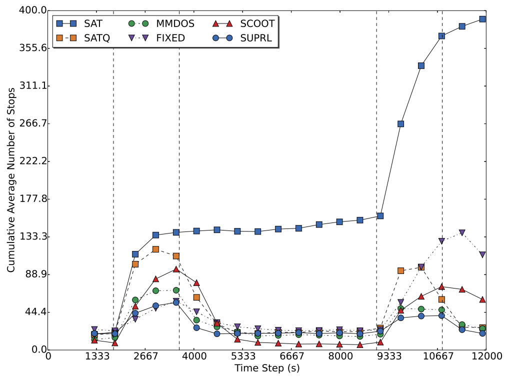
MMDOS also has the lowest CANS in *structured* and *regional* traffic, Figures 6.15b and 6.15d. Also, SCOOT has lower CANS in *structured* and *regional* traffic in comparison to SAT and SATQ. Lastly, in *constant* traffic, SCOOT has the lowest CANS. In *constant* traffic, SAT, SCOOT, FIXED and SUPRL maintain their CANS throughout the scenario, displaying little change even during the disruption. However, SATQ and MMDOS have many peaks and valleys in their CANS. Although, the CANS of MMDOS is not stable in *constant* traffic, it is lower than the CANS of SUPRL, SAT and SATQ.



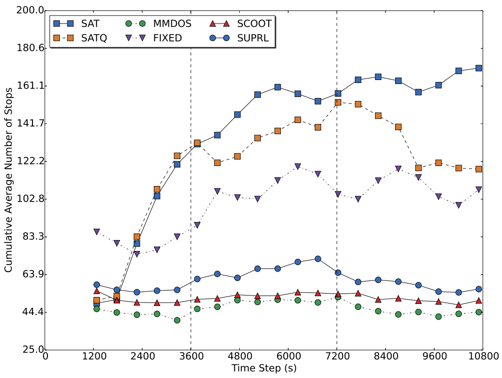
(a) Unstructured.



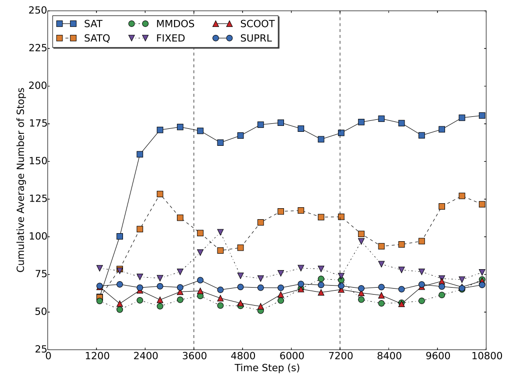
(b) Structured.



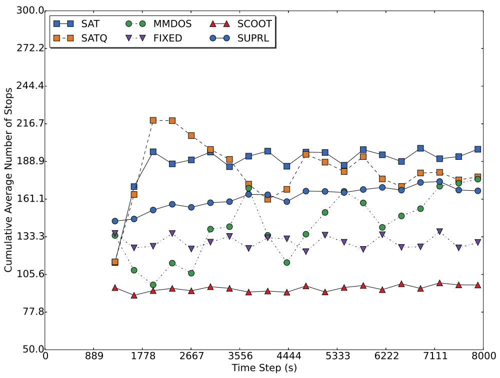
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 6.15: Cumulative average number of stops (over 30 simulations) on Phoenix map. Beginning and ending of disruptions are marked by dotted lines.

Average Travel Time (ATT) (*std.*)			
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	**229.38 (0.88)**	291.33 (0.66)	265.94 (0.86)
SAT	752.34 (75.05)	275.76 (3.49)	280.19 (16.82)
SATQ	262.9 (6.29)	244.95 (2.21)	257.82 (5.42)
MMDOS	274.62 (5.81)	273.12 (0.38)	259.2 (8.16)
SCOOT	360.86 (11)	**237.29 (3.97)**	286.92 (8.5)
SUPRL	244.13 (1.17)	246.44 (0.94)	**253.12 (1.47)**
	Unstructured	*Football*	*Directional*
FIXED	569.87 (18.05)	299.6 (5.09)	**242.97 (0.88)**
SAT	923.23 (144.23)	483.57 (22.59)	562.62 (31.47)
SATQ	564.67 (29.78)	283.9 (11.83)	270.8 (7.01)
MMDOS	584.14 (14.98)	299.94 (19.28)	270.31 (9.08)
SCOOT	**510.6 (26.52)**	392.73 (23.55)	369.13 (4.96)
SUPRL	577.7 (10)	**271.56 (5.62)**	249.16 (1.28)

Table 6.5: Average travel times (ATT) for each mechanism and traffic scenario.

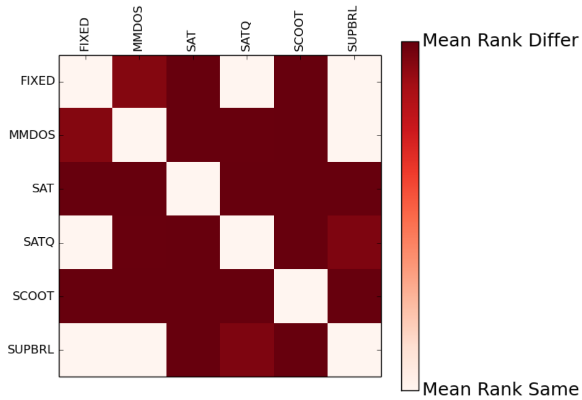
6.3 Results: Portland

This section presents the results of the experiments executed on the Portland map. This section is divided into sub-sections, covering each of the three traffic performance metrics: *average travel time* (Section 6.3.1), *traffic density* (Section 6.3.4), and *number of stops* (Section 6.3.7). The traffic control systems are evaluated in three traffic scenarios with *predictable* traffic flow (*structured*, *regional*, and *constant*) and three traffic scenarios with *unpredictable* traffic flow (*unstructured*, *football*, and *directional*). Results for ATTA (Section 6.3.3) and traffic density on a major artery (Section 6.3.6) are presented for SATQ, MMDOS, SCOOT and FIXED in the *unpredictable* traffic scenarios. SAT is not included in Sections 6.3.3 and 6.3.6 because of its poor ATT performance. Also, Sections 6.3.3 and 6.3.6 only cover the traffic scenarios with *unpredictable* traffic flow because they provide better conditions for illustrating the differences between the traffic control mechanisms. The Mann-Whitney test is used to determine statistical significance between traffic performance results. The threshold value of $p = .05$ was used to determine whether the null hypotheses (the samples were the same) was rejected. The Mann-Whitney test results are presented in a visual manner in lieu of tables to provide the same information but in a more compact manner.

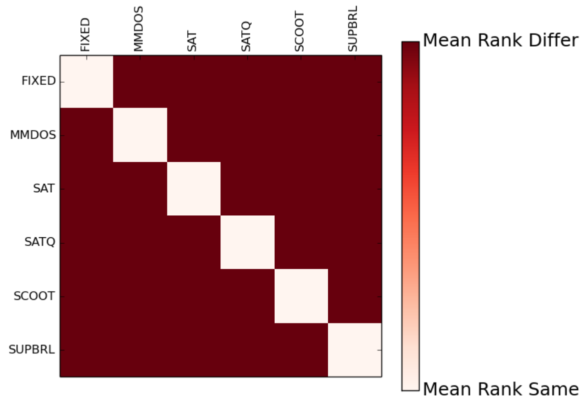
6.3.1 Travel Time (ATT)

Table 6.5 shows that on the Portland map, SCOOT has lowest ATT in the *unstructured* traffic scenario; this was not the case on the Phoenix map. Figure 6.29a shows that the difference between the ATT of SCOOT and the ATT of the other mechanisms is significant. Although SATQ and MMDOS have higher ATT than SUPRL in *unstructured* traffic, Figure 6.29a also shows that the difference between SATQ, MMDOS and SUPRL is not significant. Also, in *unstructured* traffic, SATQ has lower ATT than MMDOS. In the *football* and *directional* traffic scenario, SATQ and MMDOS have lower ATT than SCOOT. In addition, SATQ has lower ATT than MMDOS in the *football* scenario but not in the *directional* traffic scenario. SATQ and MMDOS performs as well as or worse than FIXED in *unstructured*, *football* and *directional* traffic on the Portland map. In *unstructured* and *football* traffic, the difference between the ATT of MMDOS and FIXED is not significant and in *unstructured* traffic SATQ is not significantly different from FIXED (see Figure 6.29). In contrast, on the Phoenix map SATQ and MMDOS have lower ATT than FIXED in *unstructured* and *football* traffic (in *directional* traffic on the Phoenix map MMDOS has lower ATT than FIXED but not SATQ). SAT has the highest ATT in all three traffic scenarios with *unpredictable* traffic flow. Lastly, the difference between the market-based approaches and the benchmarks in *directional* traffic is significant (see Figure 6.29).

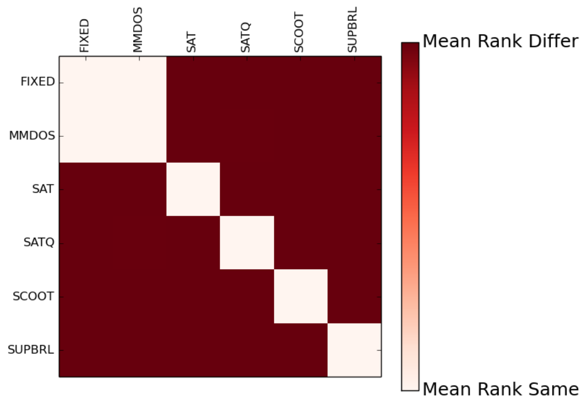
In *structured* and *constant* traffic, SATQ and MMDOS have lower ATT than SCOOT but both have higher ATT than SUPRL. In *structured* traffic, SATQ and MMDOS have greater ATT than FIXED but in *constant* traffic, SATQ and MMDOS have a slight edge over FIXED. In *regional* traffic, SATQ has lower ATT than SUPRL, MMDOS and FIXED, however, it does not have lower ATT than SCOOT. SAT has the highest ATT in *structured* and *constant* traffic but not in *regional* traffic. The difference between the market-based approaches and the benchmarks in all three traffic scenarios with *predictable* traffic flow is significant.



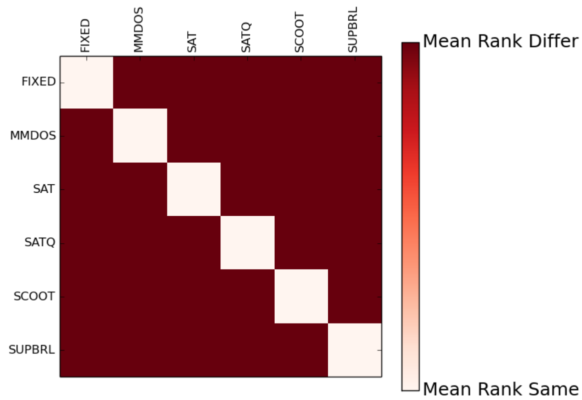
(a) Unstructured.



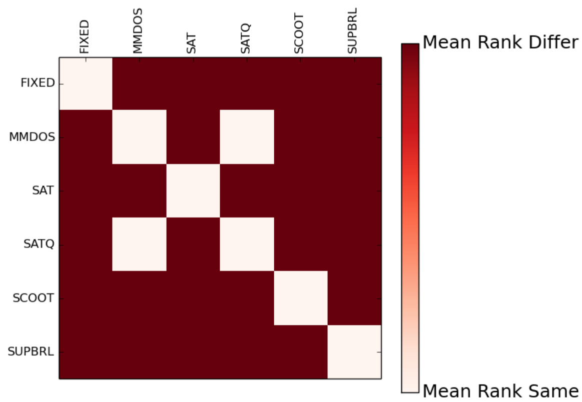
(b) Structured.



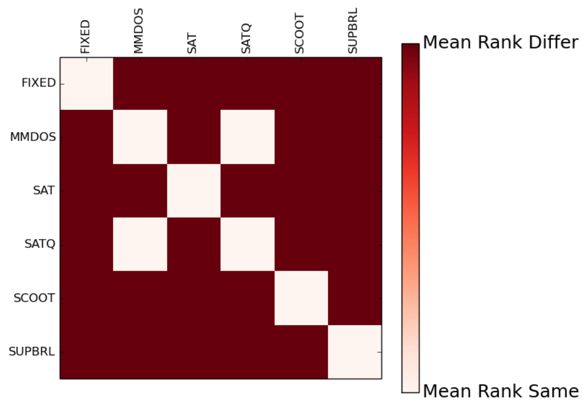
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 6.16: Visual representation of two-sample Mann-Whitney test conducted on ATT (Portland map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

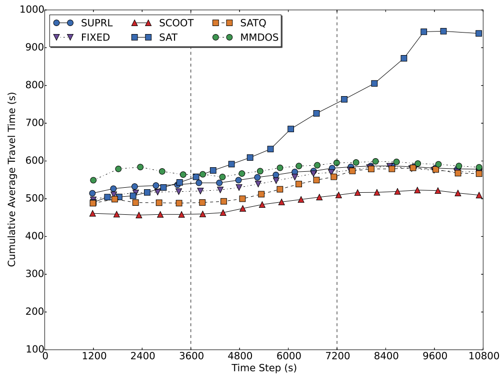
6.3.2 Cumulative Average Travel Time (CATT)

On the Portland map with *unstructured* traffic, SATQ has the second lowest CATT (SCOOT has the lowest) (see Figure 6.17a). During the disruption, the CATT of SATQ rises and eventually reaches the same level as FIXED and SUPRL but does not surpass it either. Figure 6.17a also shows that initially, MMDOS has the highest CATT, however, at the start of the disruption, SAT surpasses MMDOS and all the other mechanisms. After the *unstructured* traffic disruption, MMDOS has similar CATT to SATQ, FIXED, and SUPRL. SAT has a sharp increase in CATT during the disruption in comparison to the other mechanisms, this sharp increase in CATT occurs on both maps.

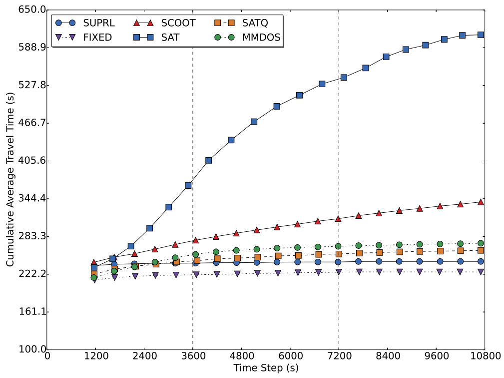
In the *football* scenario, all the mechanisms display nearly the same average travel time during (and before) the first disruption, see Figure 6.17c. During the football match the mechanisms showed a much slower recovery, i.e., peak average travel times occurs farther into the match than on the Phoenix map. Although SATQ has a higher peak CATT than MMDOS during the match, both mechanisms have similar CATT towards the end of the match and onward, see Figure 6.17c. Lastly, the CATT of SCOOT and SAT begins to rise at the start of the first disruption and continues to do so throughout the football match. However, SCOOT's CATT plateaus at the end of the match but the CATT of SAT continues to increase. Lastly, Figure 6.17c shows that from the beginning of the match until the end of the scenario, FIXED has the lowest CATT.

FIXED also has lower CATT in *directional* traffic, see Figure 6.17e. Also, in *directional* traffic, MMDOS and SATQ have similar CATT. The CATT results in *structured* traffic resembles the CATT in *directional* traffic, see Figure 6.17b. In *directional* and *structured* traffic, only SAT and SCOOT display an increase in CATT during the disruptions. In *regional* traffic, Figure 6.17d, the market-based traffic control systems have lower CATT than FIXED but higher than SCOOT. Also, in the *regional* traffic scenario, SAT outperforms MMDOS and SATQ performs similar to SUPRL. Figure 6.17f, shows that initially, MMDOS performs similar to SCOOT and SAT performs similar to SATQ. However, as the *constant* traffic scenario progresses, MMDOS and SATQ have nearly identical CATT; both mechanisms have lower CATT than SCOOT and FIXED. Additionally, the CATT of SAT and SCOOT increases throughout the *constant* traffic scenario. On the Phoenix map, SCOOT has the lowest CATT in *directional* traffic and the three traffic scenarios with *predictable* traffic flow.

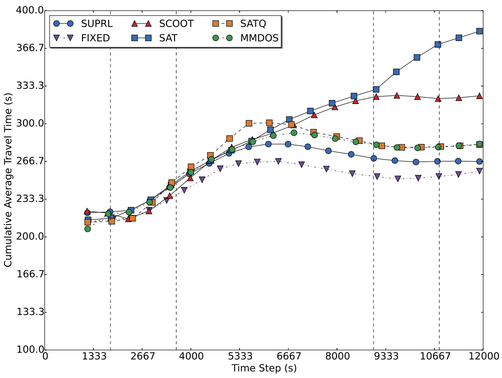
In the *regional* traffic scenario, on the Portland map, the mechanisms show little change in CATT during the disruption, see Figure 6.17d. Lastly, in *constant* traffic, Figure 6.17f, shows that initially SCOOT, SAT and SATQ have lower CATT than SUPRL and near the middle of the scenario surpass the CATT of SUPRL. However, the CATT of FIXED and SUPRL show little change with *constant* traffic.



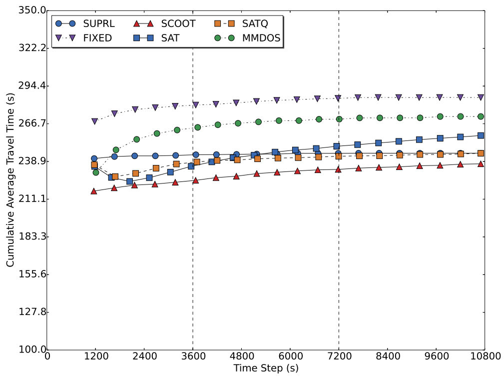
(a) Unstructured.



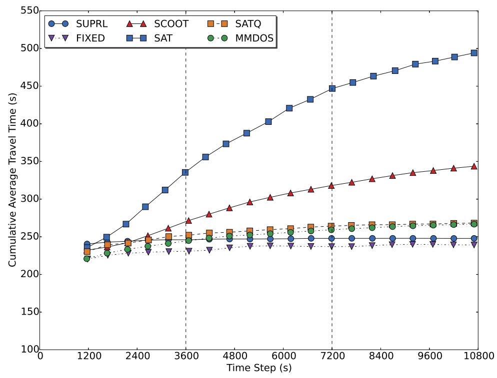
(b) Structured.



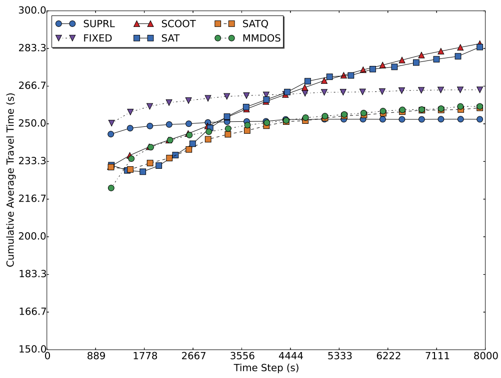
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 6.17: Cumulative average travel times (over 30 simulations) on Portland map. Beginning and ending of disruptions are marked by dotted lines.

6.3.3 Average Travel Time on Arrival (ATTA)

A closer examination of travel times, more specifically, the average travel times of vehicle that have completed their journey at each time step reveals that with *unstructured* traffic, the ATTA of SATQ is similar to SCOOT's ATTA, see Figure 6.18. Figure 6.18 also shows that SATQ has a bit more vehicles that have higher ATTA than SCOOT, especially during the disruption period. SATQ has more vehicles with ATTA below 400 seconds than SUPRL in *unstructured* traffic. Additionally, in *unstructured* traffic, MMDOS has similar ATTA to SUPRL, see Figure 6.19.

Figures 6.20 and 6.21 show that in the *football* scenario, ATTA varied far more on the Portland map than it did on the Phoenix map. SATQ in the *football* scenario, has ATTA closer to SUPRL than SCOOT. For example, both SATQ and SUPRL have vehicles with ATTA higher than 500 seconds prior to and during the initial portion of the first disruption. MMDOS displays a similar pattern of ATTA, thus, it too is more similar to SUPRL than SCOOT.

However, in *directional* traffic, Figure 6.22, SATQ has similar ATTA patterns to SCOOT. In *directional* traffic, SATQ and SCOOT show a greater distribution of ATTA than SUPRL, on the Portland map compared to *directional* traffic on the Phoenix map. On the Phoenix map, in *directional* traffic, SATQ forms two clusters of ATTA. Figure 6.23 shows that in *directional* traffic, the ATTA of MMDOS is similar to that of SUPRL. The ATTA of SUPRL and MMDOS is consistent on both maps in *directional* traffic. On both maps, the ATTA of SUPRL and MMDOS fall within a narrow range throughout the scenario. However, in *directional* traffic, on the Portland map, SUPRL has more vehicles with ATTA lower than 250 seconds.

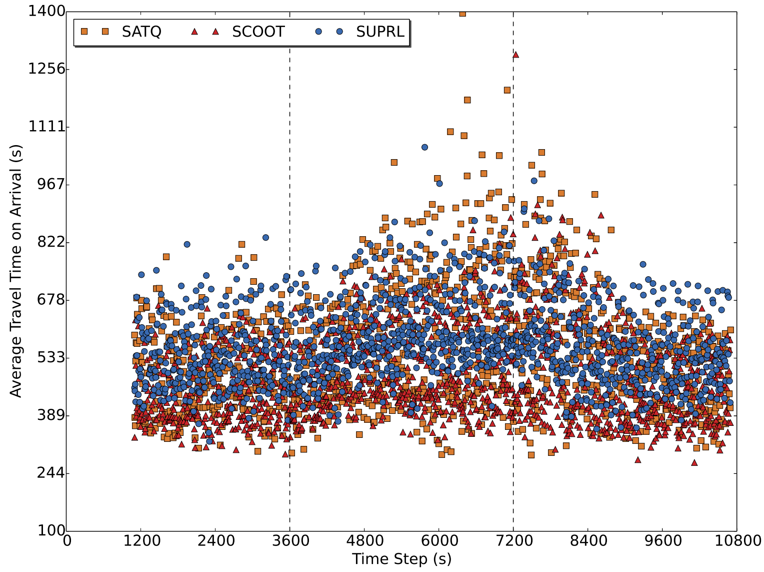


FIGURE 6.18: The graph shows the average travel times of vehicles that have completed their journey at each time step. (over 30 simulations) in *unstructured* traffic on the Portland map.

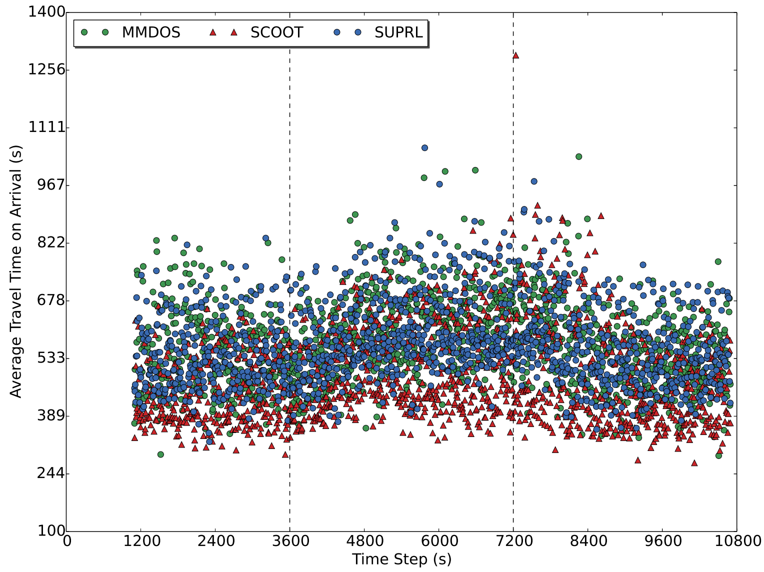


FIGURE 6.19: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *unstructured* traffic on the Portland map.

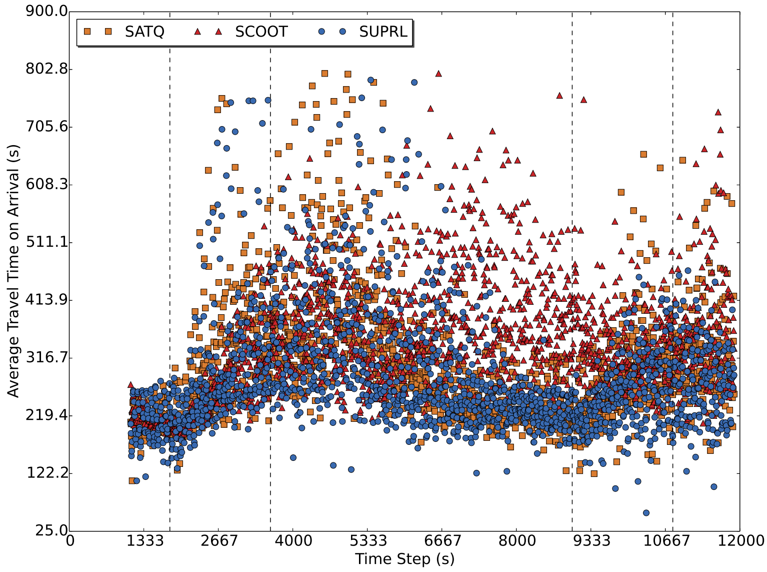


FIGURE 6.20: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *football* traffic on the Portland map.

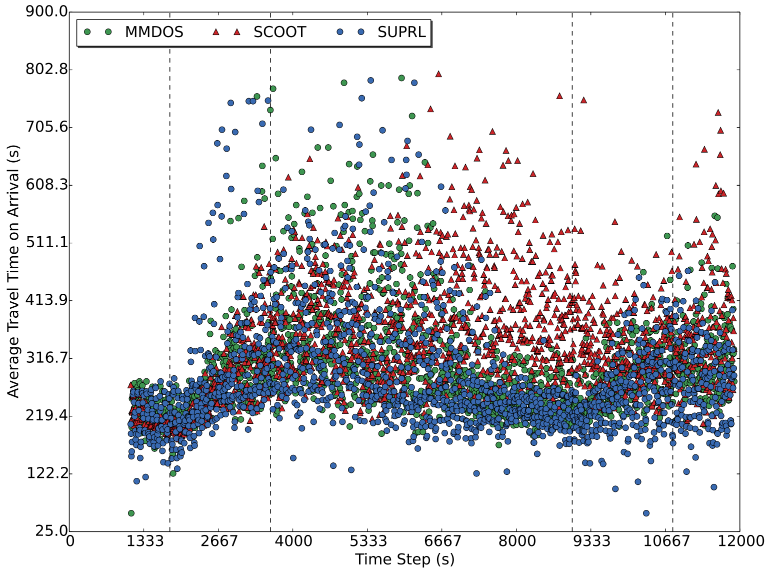


FIGURE 6.21: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *football* traffic on the Portland map.

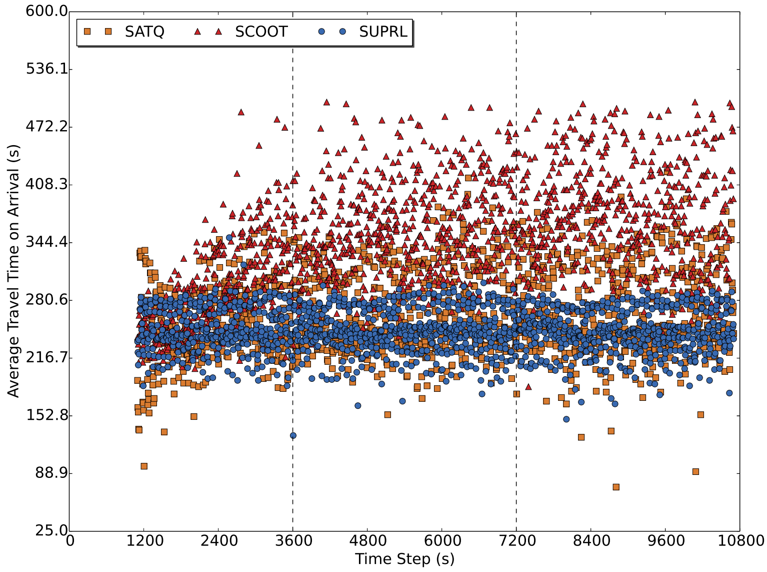


FIGURE 6.22: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *directional* traffic on the Portland map.

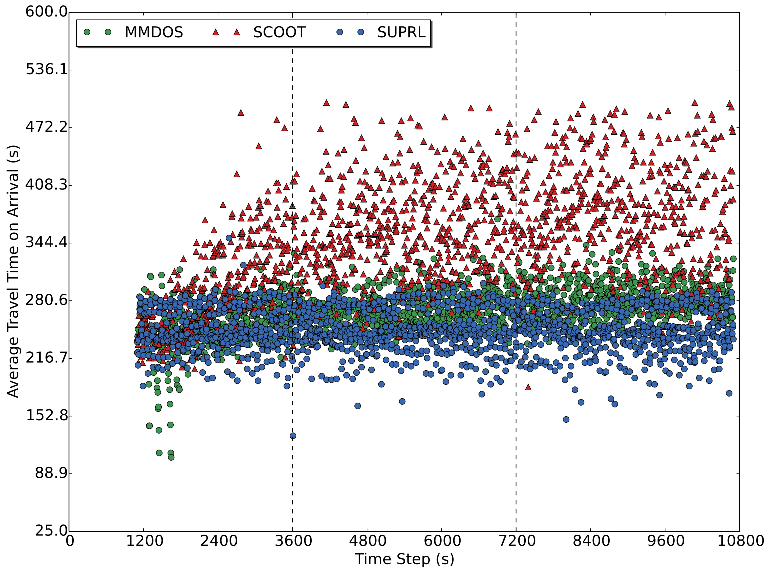


FIGURE 6.23: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *directional* traffic on the Portland map.

6.3.4 Density (ATD)

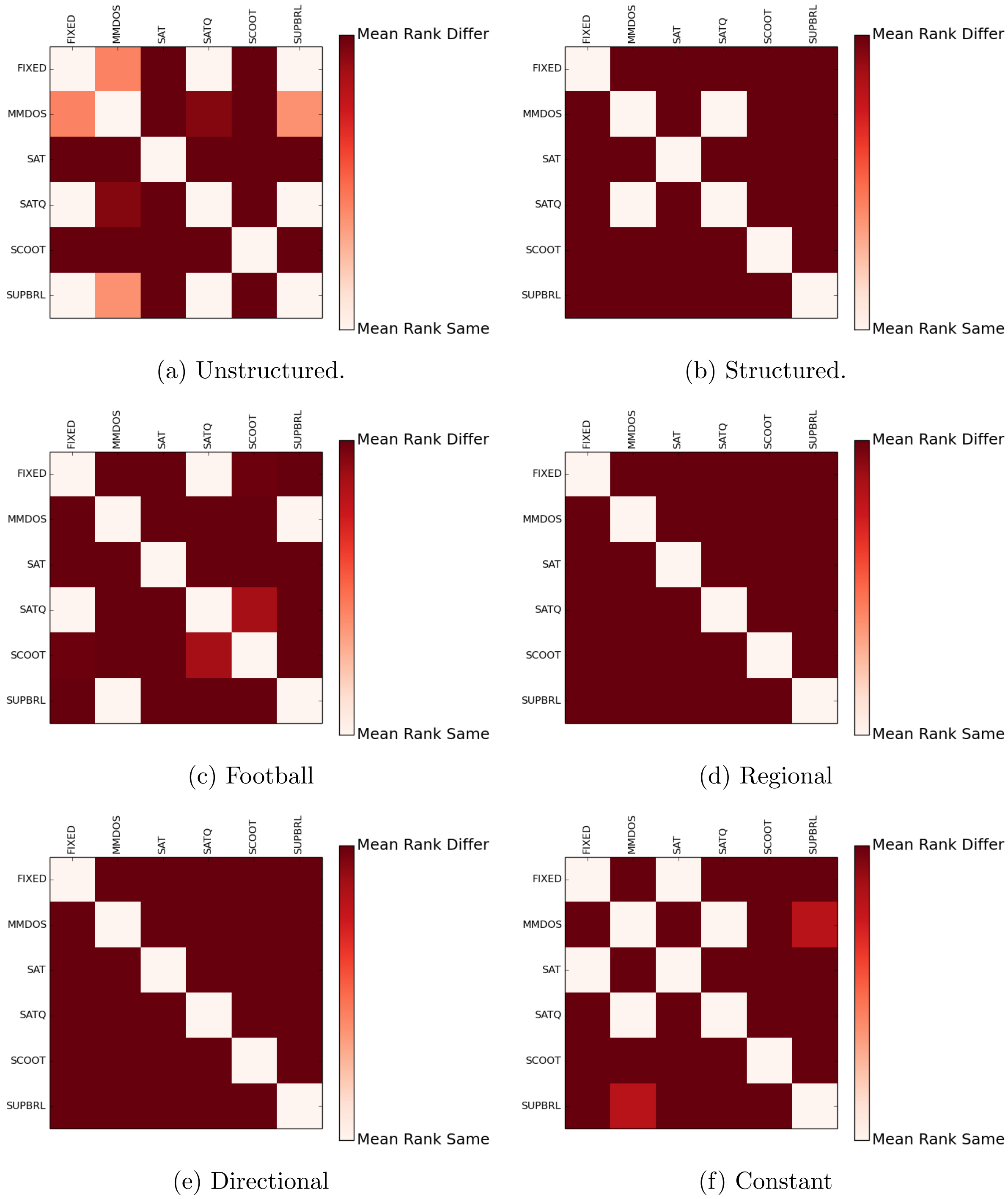
Average Traffic Density (ATD) (*std.*)			
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	**12.42 (0.21)**	20.46 (0.2)	20.69 (0.16)
SAT	26.07 (2.58)	**16.3 (0.53)**	20.92 (1.48)
SATQ	15.57 (0.41)	17.99 (0.24)	19.96 (0.48)
MMDOS	15.39 (0.57)	20.25 (0.2)	19.8 (0.63)
SCOOT	18.24 (0.58)	16.98 (0.39)	22.66 (0.75)
SUPRL	13.94 (0.23)	17.39 (0.2)	**19.52 (0.19)**
	Unstructured	*Football*	*Directional*
FIXED	2.76 (0.14)	9.54 (0.15)	**14.03 (0.18)**
SAT	13.13 (0.69)	12 (0.55)	20.54 (1.27)
SATQ	2.75 (0.18)	9.61 (0.36)	16.45 (0.47)
MMDOS	2.84 (0.12)	**8.92 (0.29)**	15.43 (0.71)
SCOOT	**2.43 (0.16)**	9.94 (0.52)	19.24 (0.38)
SUPRL	2.78 (0.11)	8.96 (0.26)	14.76 (0.13)

TABLE 6.6: Average traffic density (ATD) for each mechanism and traffic scenario.

Table 6.6 contains the ATD for each mechanism in the six traffic scenarios. Table 6.6 shows that in general the difference in ATD of the mechanisms is small, especially in *unstructured* traffic. Nonetheless, in *unstructured* traffic, SCOOT has the lowest ATD. Also, in *unstructured* traffic, MMDOS and SATQ are not significantly different from FIXED and SUPRL, see Figure 6.24a.

In the *football* scenario, MMDOS has the lowest ATD, however, it is not significantly different from the ATD of SUPRL, see Figure 6.24c. Figure 6.24c also shows that in the *football* traffic scenario, the ATD of SATQ is not significantly different from FIXED and SCOOT. In *directional* traffic, SATQ and MMDOS have lower ATD than SCOOT but not SUPRL; FIXED has the lowest ATD in *directional* traffic. MMDOS has lower ATD than SATQ in *football* and *directional* traffic, in both scenarios the difference is significant, see Figure 6.24. SAT has the highest ATD in *unstructured*, *football* and *directional* traffic and the difference between SAT and the other mechanisms is also significant, see Figure 6.24.

SATQ and MMDOS have lower ATD than SCOOT in *structured* and *constant* traffic. Figure 6.24 shows that in both traffic scenarios, the difference between the market-based mechanisms and SCOOT is significant. Figure 6.24 also shows that in *constant* traffic, MMDOS and SUPRL are not significantly different. Also, FIXED and SUPRL have the lowest ATD in *structured* and *constant* traffic, respectively. In *regional* traffic, SAT has the lowest ATD, however, SAT does have the highest ATD in *structured* and *constant* traffic. Also, in *regional* traffic, MMDOS and SATQ have higher ATD than SCOOT and SUPRL. The difference between the market-based traffic control systems and the benchmarks is significant in *regional* traffic, see Figure 6.24d.



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

(e) Directional

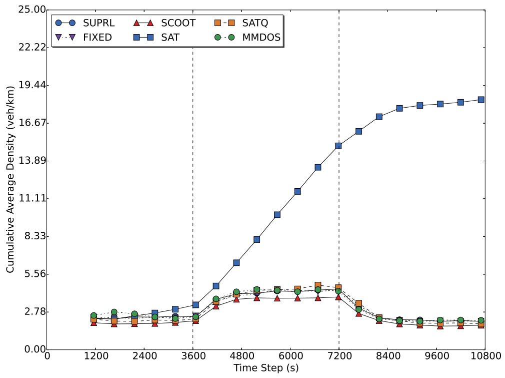
(f) Constant

FIGURE 6.24: Visual representation of two-sample Mann-Whitney test conducted on ATD (Portland map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

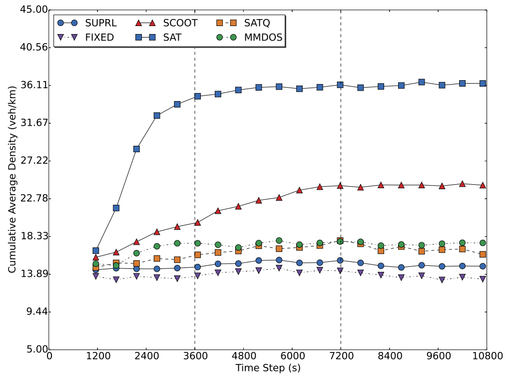
6.3.5 Cumulative Average Density (CAD)

In the *unstructured* traffic scenario, all the mechanisms, excluding SAT, have nearly the same CAD; this is not the case on the Phoenix map, see Figure 6.25a. However, on both maps, SAT displays higher CAD levels than the other traffic control systems. Although initially all the mechanisms behave in a similar manner in the *football* scenario, differences in CAD begin to appear during the football match, see Figure 6.25c. In the middle of the match when MMDOS, SUPRL, SATQ, FIXED begin to recover from the first disruption while SAT and SCOOT maintain their CAD levels. On the second disruption, the CAD of all of the mechanism increases, however, the CAD of MMDOS, SATQ and SUPRL quickly plateaus, unlike the CAD of SCOOT, FIXED and SATQ. In *directional* traffic, Figure 6.25e, SATQ and MMDOS maintain similar CAD while SAT and SCOOT have much higher CAD levels.

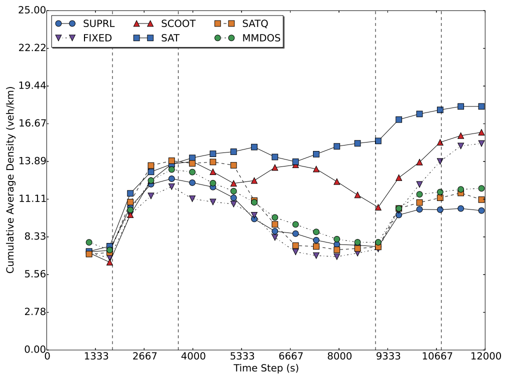
The CAD of the mechanism in *structured* traffic, mirror *directional* traffic. In both scenarios, SATQ and MMDOS have similar CAD levels and SCOOT and SAT have higher CAD than all the other mechanisms. Additionally, in *directional* and *structured* traffic, FIXED have the lowest CAD. However, in *regional* traffic, FIXED has the highest CAD, see Figure 6.25d. In *regional* traffic, SATQ has lower CAD than SATQ and MMDOS and during certain period performs similar to SCOOT and SUPRL. Also, in *regional* traffic, MMDOS outperforms FIXED but has higher CAD than the other two market-based systems, SCOOT and SUPRL. In *constant* traffic, Figure 6.25f, SATQ and MMDOS have similar CAD. Additionally, MMDOS has the lowest CAD and displays similar CAD to SUPRL during certain periods in *constant*. Lastly, in *constant* traffic SCOOT has the highest CAD. This is different from the experiments on the Phoenix map where SCOOT has the lowest CAD, Figure 6.25f, in *constant* traffic.



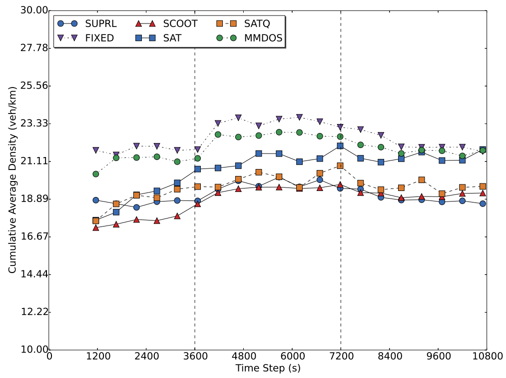
(a) Unstructured.



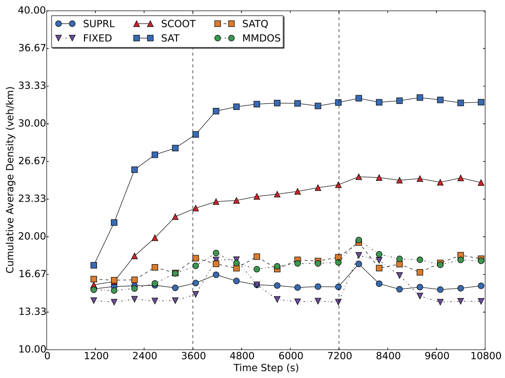
(b) Structured.



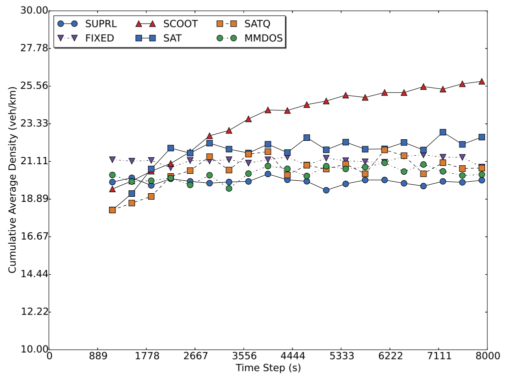
(c) Football



(d) Regional



(e) Directional



(f) Constant

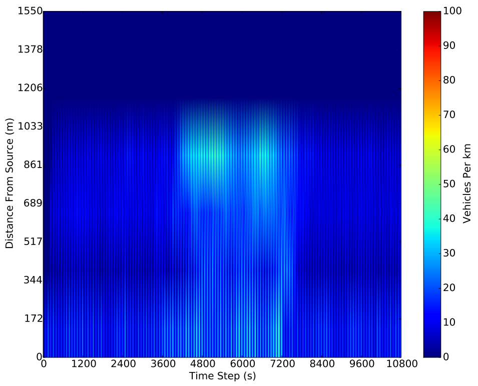
FIGURE 6.25: Cumulative average density (over 30 simulations) on Portland map. Beginning and ending of disruptions are marked by dotted lines.

6.3.6 Density on Major Artery

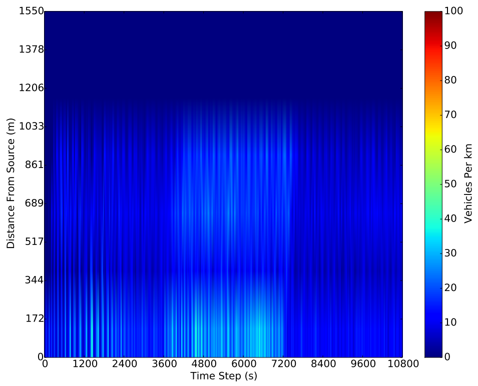
This section discusses the traffic density on a single artery on the Portland map, from the left, the second south-to-north lane of traffic. Figures 8.20a shows that under SATQ, traffic density with *unstructured* traffic on the second artery is very low with an accumulation of traffic near the end of the artery. In the same scenario, SUPRL, MMDOS, and FIXED, shown in Figure 8.20, have little traffic on the second artery as well. SCOOT has the lowest traffic density levels on the second artery in comparison to SATQ, SAT, and MMDOS, see Figure 8.20d. Lastly, Figure 8.20e, shows that SAT has the highest traffic density on the second artery than all of the other mechanisms.

Figure 8.21 shows that in the *football* scenario, SATQ has a period with higher traffic density, however, that period is short and less intense than traffic density on the second artery with SCOOT, SATQ or FIXED. Figure 8.21 also shows that, in the *football* scenario, traffic density on the second artery with MMDOS is similar to SUPRL. MMDOS and SUPRL have higher traffic density on the third road segment of the second artery during and shortly after the first disruption. Lastly, SATQ has higher traffic density on the second artery much later in the scenario than the other mecahnisms.

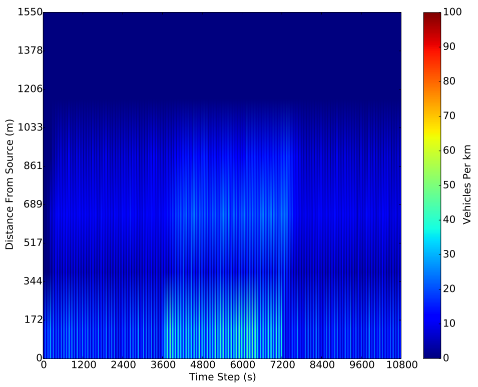
In *directional* traffic, all of the mechanisms display a similar pattern of traffic density along the fourth artery, see Figure 8.22. In the *directional* traffic scenario on the Portland map, all of the mechanism have long vehicle queues on the first road segment. Also, MMDOS, SUPRL and FIXED have greater traffic density on the first road segment during the disruption than SATQ and SCOOT. Figure 8.22b shows that although the intensity of the traffic flow heading north is reduced, vehicle queue remain on the first road segment with MMDOS, similar to traffic on the first road segment with SUPRL and FIXED. The traffic density on the fourth road segment with SATQ is similar to traffic density with SAT, however, it appears SAT has lower traffic density downstream than SATQ.



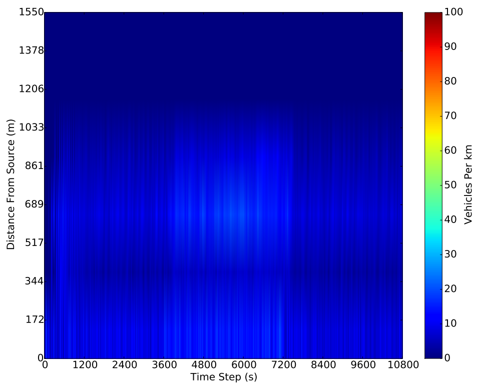
(a) SATQ



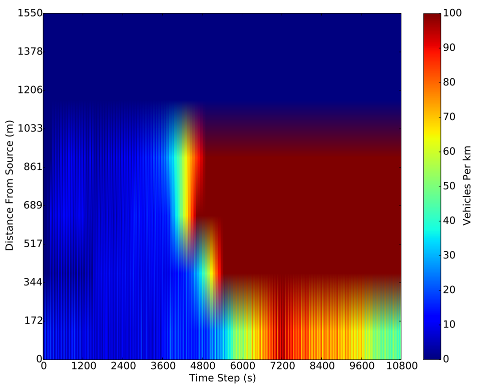
(b) MMDOS.



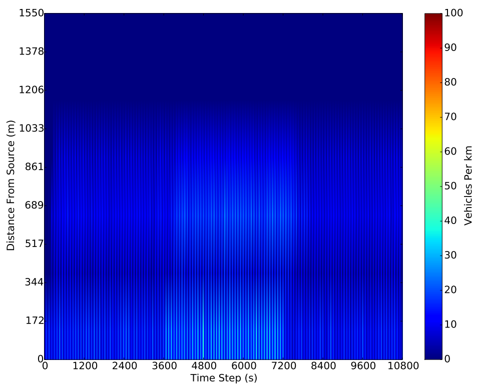
(c) SUPRL



(d) SCOOT

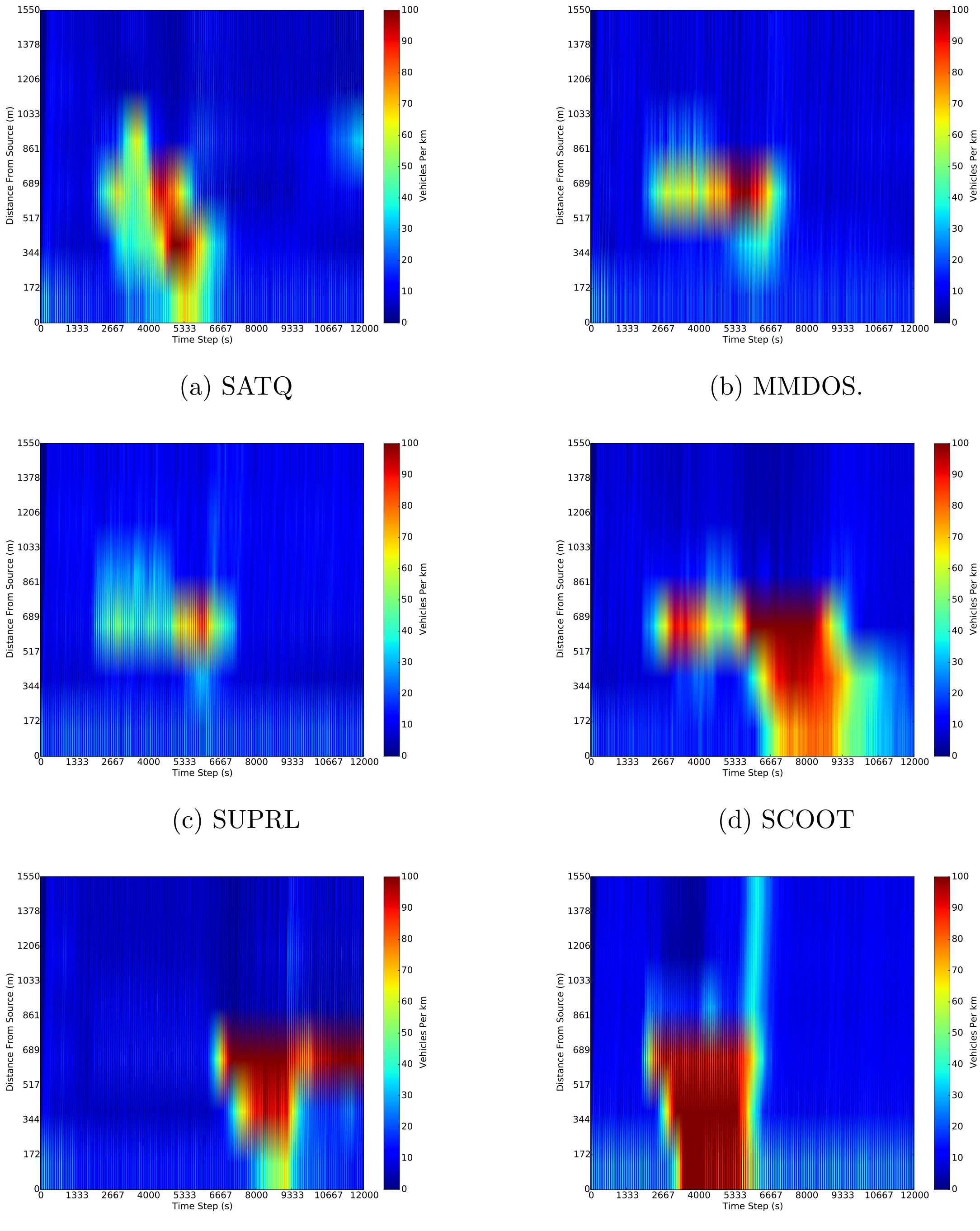


(e) SAT



(f) FIXED

FIGURE 6.26: Traffic density on major artery in *unstructured* traffic on Portland map.



(a) SATQ

(b) MMDOS.

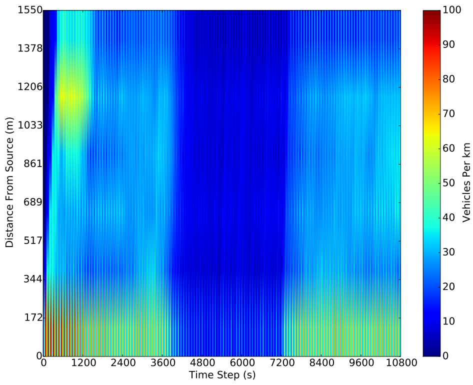
(c) SUPRL

(d) SCOOT

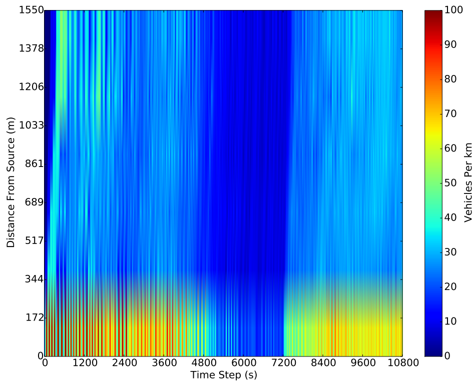
(e) SAT

(f) FIXED

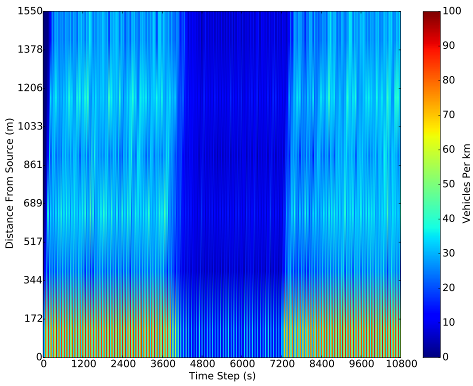
FIGURE 6.27: Traffic density on major artery in *football* traffic on Portland map.



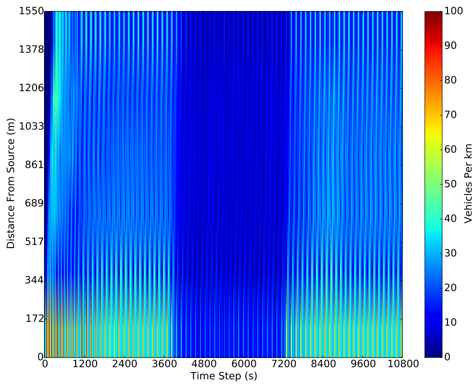
(a) SATQ



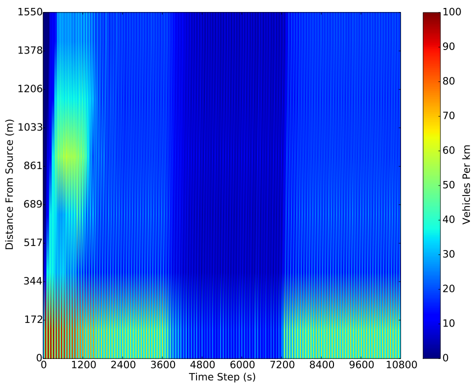
(b) MMDOS.



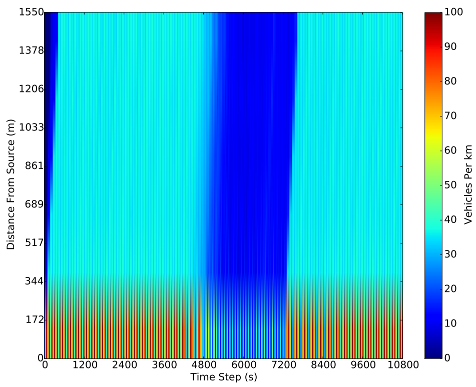
(c) SUPRL



(d) SCOOT



(e) SAT



(f) FIXED

Figure 6.28: Traffic density on major artery in *directional* traffic in Portland map.

Average Number of Stops (ANS) (*std.*)			
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	**161.13 (2.71)**	280.5 (2.71)	279.97 (2.54)
SAT	566.47 (63.9)	231.63 (9.1)	311.67 (37.65)
SATQ	220.23 (8.25)	217.6 (4.54)	262.27 (10.19)
MMDOS	222.6 (10.41)	203.8 (2.41)	**226.03 (8.29)**
SCOOT	319.8 (13.78)	203.77 (7.74)	347.6 (19.31)
SUPRL	185.33 (3.18)	**201.1 (2.68)**	246 (3.45)
	Unstructured	*Football*	*Directional*
FIXED	28.83 (2.45)	143.4 (3.61)	**187.1 (2.5)**
SAT	366.43 (16.35)	251.43 (13.66)	416.83 (29.9)
SATQ	31.27 (3.59)	136.07 (8.83)	232.2 (9.13)
MMDOS	31.8 (2.01)	127.57 (9.9)	209.53 (12.27)
SCOOT	**19.27 (2.27)**	153.77 (13.37)	344.93 (7.87)
SUPRL	29.63 (1.52)	**122.73 (4.89)**	194.37 (2.34)

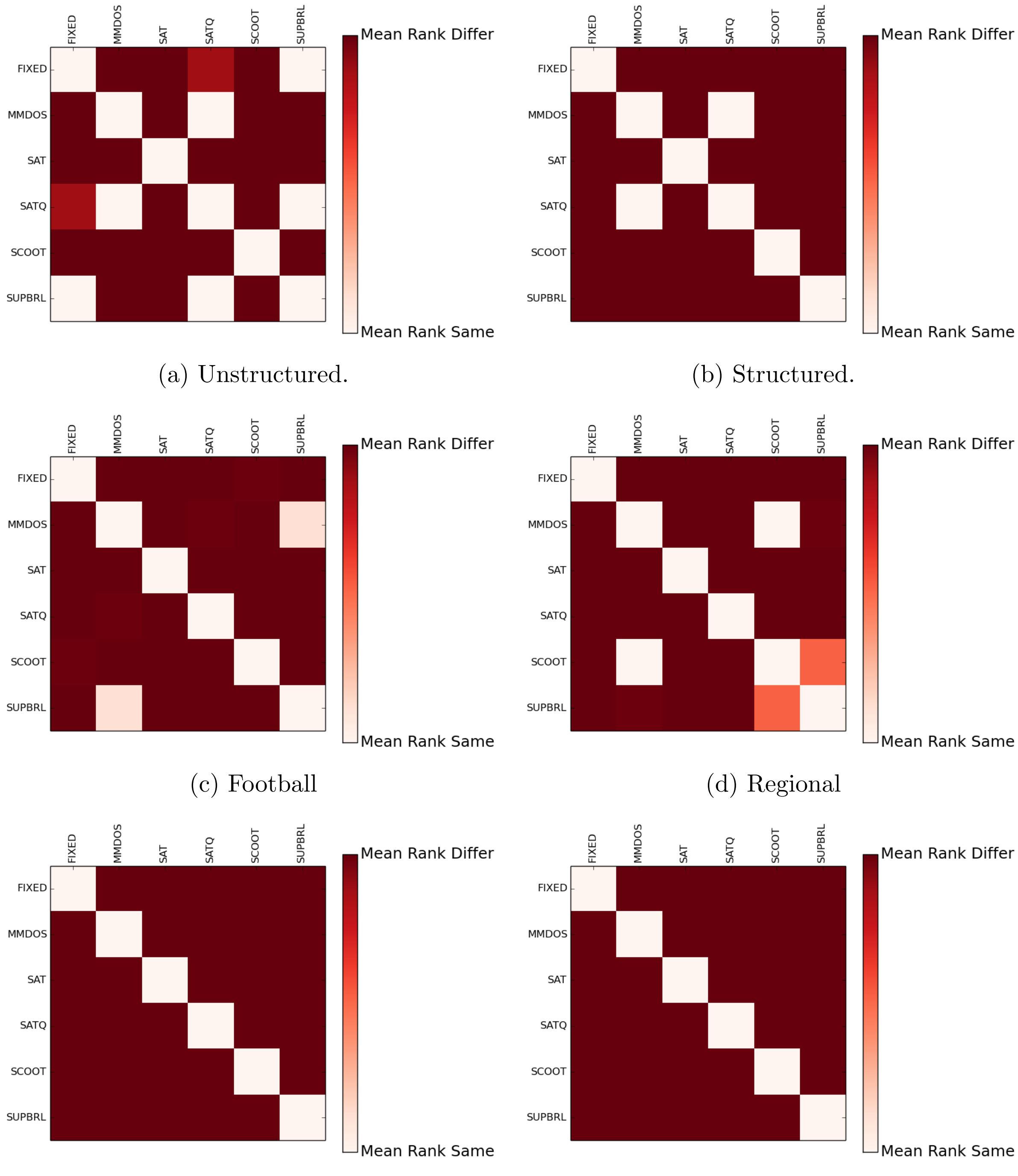
Table 6.7: Average *number of stops* (ANS) for each mechanism and traffic scenario.

6.3.7 Vehicle Stops (ANS)

On the Portland map SCOOT has the lowest ANS in the *unstructured* traffic scenario, Figure 6.29a shows that this is statistically significant. Statistically, the performance of FIXED, SATQ and SUPRL do not differ in *unstructured* traffic, see Figure 6.29a. In *unstructured* traffic, the ANS of SATQ is higher than but close to that of FIXED, SUPRL and MMDOS. Also, in *unstructured* traffic, MMDOS which has higher ANS than SCOOT, SUPRL and FIXED, is not significantly different from SATQ. In the *football* scenario SATQ has lower ANS than SAT, SCOOT and FIXED. However, MMDOS has lower ANS than all of the other mechanisms, except SUPRL, in the *football* scenario. Figure 6.29c shows that, although, SUPRL has the overall lowest ANS in the *football* scenario it is not significantly different from MMDOS. MMDOS and SATQ have lower ANS than SAT and SCOOT in *directional* traffic (FIXED has the lowest ANS in this scenario). Figure 6.29e, shows that in *directional* traffic, the difference between the market-based traffic controllers and the benchmarks is significant.

On the Portland map SATQ also has lower ANS than SCOOT in some of the traffic scenarios with *predictable* traffic. Table 6.7 shows that in *structured* and *constant* traffic SATQ has lower ANS than SCOOT. MMDOS also has lower ANS than SCOOT in *structured* and *constant* traffic In addition, in *constant* traffic, MMDOS has lower ANS than SUPRL and SAT has lower ANS than SCOOT. In *regional* traffic, SAT, SATQ and MMDOS all have lower ANS than FIXED but not SCOOT and SUPRL. However, in the *regional* traffic scenario, MMDOS and SCOOT are close in terms of ANS. Figure 6.29

shows that in the traffic scenarios with *predictable* traffic, the difference between the market-based mechanisms and the benchmarks is significant.



(a) Unstructured. (b) Structured.

(c) Football (d) Regional

(e) Directional (f) Constant

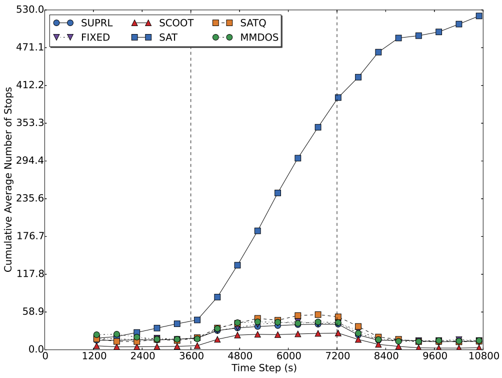
FIGURE 6.29: Visual representation of two-sample Mann-Whitney test conducted on ANS (Portland map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

6.3.8 Cumulative Average Number of Stops (CANS)

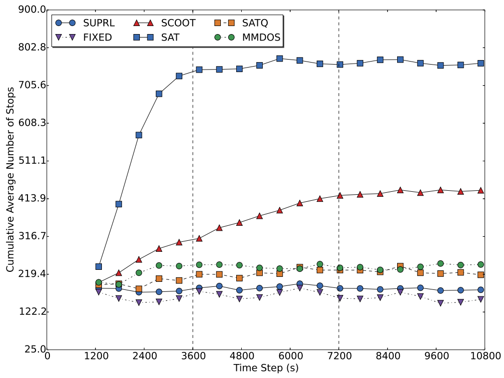
In *unstructured* traffic, excluding SAT, the mechanisms have far fewer stops compared to the other scenarios. Figure 6.30a shows that on the Portland map, in *unstructured* traffic, MMDOS, SUPRL and SATQ have similar CANS. Figure 6.30a also shows how well SCOOT performs in the *unstructured* traffic scenario; SCOOT maintains low CANS throughout the entire scenario. Lastly, in *unstructured* traffic, SAT has a sharp increase in CANS which begins during the disruption and continues even after the disruption ends.

In the *football* scenario, SATQ and MMDOS performs just as well as the benchmarks during the first disruption, i.e., all the mechanisms experience a similar increase in CANS, see Figure 6.30c. However, during the *football* match, SATQ and MMDOS recovers from the initial increase quicker than SCOOT. SATQ and MMDOS also, continues to outperform SCOOT during the second disruption as well. SAT has the highest CANS in during the entire *football* scenario.

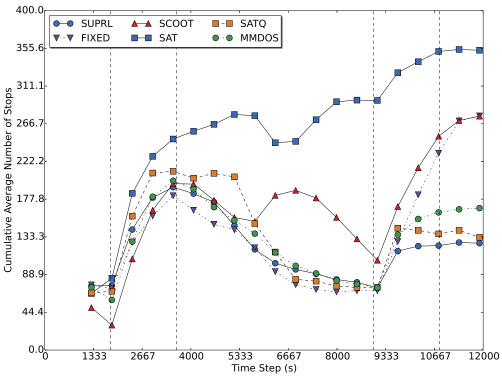
SATQ and MMDOS also outperforms SCOOT in *structured* and *directional* traffic, see Figure 6.30. In *structured* and *directional* traffic, on the Portland map, the CANS of SCOOT begins to increase prior to the disruption and does not plataeu until after the disruption ends. Although SCOOT has a poor performance in both scenarios, the CANS of SAT surpasses SCOOT in both traffic scenarios on the Portland map. In *regional* traffic, SATQ and SCOOT have similar CANS, although, initially SCOOT has lower CANS than all of the mechanisms, see Figure 6.30d. Also in *regional* traffic, MMDOS and SUPRL have similar CANS. Lastly, in *regional* traffic, SAT has higher CANS than all of the mechanisms except FIXED. In *constant* traffic, MMDOS has the lowest CANS, although, as certain points in the scenario the CANS of SUPRL is as low as MMDOS, see Figure 6.30f. Also, in *constant* traffic, SATQ and FIXED have similar CANS. Lastly, in *constant* traffic, shown in Figure 6.30f, SCOOT has the highest CANS surpassing even SAT.



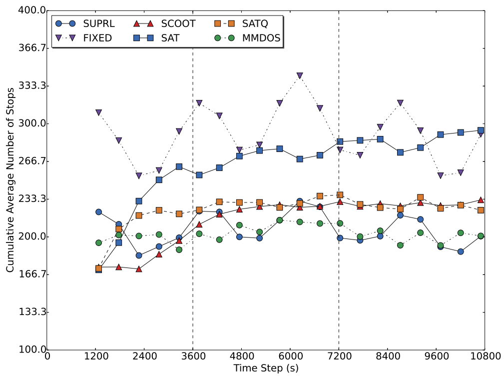
(a) Unstructured.



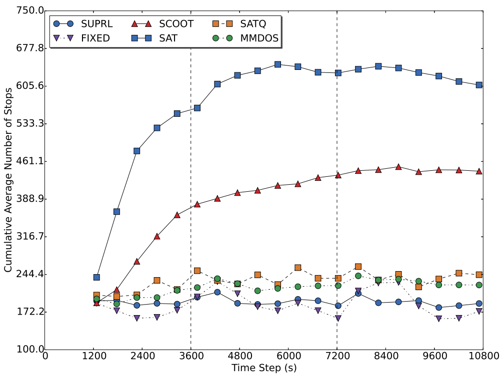
(b) Structured.



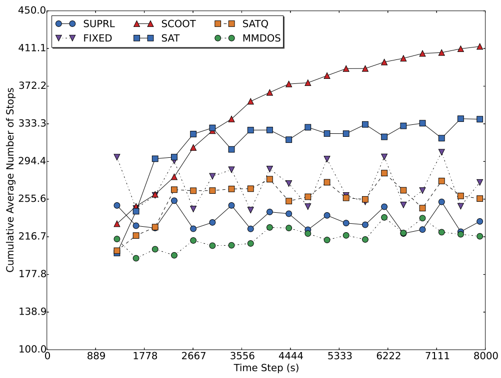
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 6.30: Cumulative average number of stops (over 30 simulations) on Portland map. Beginning and ending of disruptions are marked by dotted lines.

6.4 Summary

This section presents a summary of my findings for SAT, SATQ and MMDOS. The ATT, ATD and ANS results discussed in this section are aggregated across both maps.

6.4.1 ATT

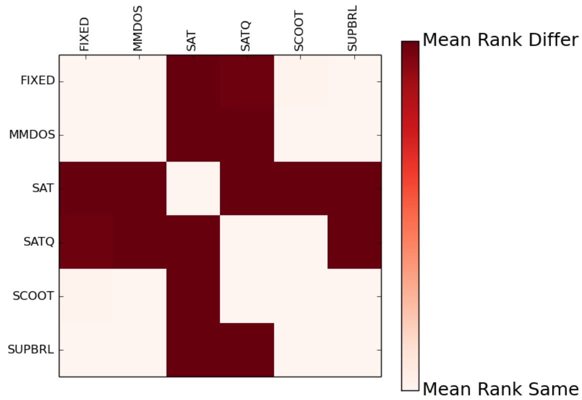
Average Travel Time (ATT) (*std.*)			
		Traffic Pattern	
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	**197.75 (31.92)**	237.7 (54.1)	225.27 (41.02)
MMDOS	212.44 (62.86)	208.7 (64.96)	225.02 (35.92)
SAT	564.02 (197.18)	252.19 (24.18)	248.45 (34.21)
SATQ	223.2 (40.35)	217.75 (27.56)	233.74 (24.78)
SCOOT	252.83 (109.24)	**183.36 (54.52)**	**215.81 (72)**
SUPRL	201.8 (42.7)	195.23 (51.65)	229.62 (24.35)
	Unstructured	*Football*	*Directional*
FIXED	839.34 (296.72)	245.25 (55.66)	208.08 (35.21)
MMDOS	651.03 (88.08)	227.11 (74.75)	215.24 (55.95)
SAT	1336.09 (664.06)	510.5 (32.39)	461.64 (104.4)
SATQ	**584.74 (32.48)**	229.2 (55.98)	232.33 (39.24)
SCOOT	870.98 (446.74)	288.77 (106.27)	258.03 (112.15)
SUPRL	716.68 (150.72)	**207.16 (65.13)**	**204.78 (44.77)**

Table 6.8: Average travel times (ATT) for each mechanism and traffic scenario.

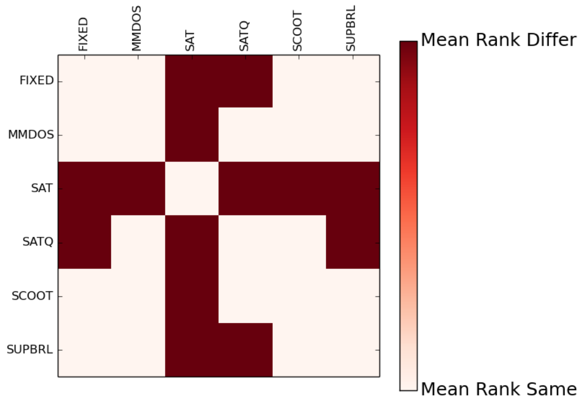
Table 6.8 contains the ATT across both maps for the mechanisms evaluated in this chapter. In unstructured traffic, SATQ has the lowest ATT; Figure 6.31 shows that the ATT of SATQ is significantly different from FIXED and SUPRL but not SCOOT. In all of the traffic scenarios, except, unstructured traffic, MMDOS has lower ATT than SAT and SATQ. Additionally, the ATT is significantly different from SAT in every traffic scenario. However, the difference between MMDOS and SATQ is only significant in unstructured traffic. Although in the majority of the traffic scenarios the market-based approaches fail to outperform all three benchmarks, SATQ and MMDOS does perform well in comparison to SCOOT. In unstructured, football and directional and structured traffic, MMDOS and SATQ outperform SCOOT.

In all of the traffic scenarios, excluding *unstructured* traffic, one of the benchmarks has the lowest ATT. In the football and directional traffic scenarios, SUPRL has the lowest ATT (the only significant difference is between the ATT of SUPRL and MMDOS in *directional* traffic, see Figure 6.31). In *structured* traffic FIXED has the lowest ATT; however, the ATT of FIXED in *structured* traffic is not significantly different from MMDOS. Finally, in *regional* and *constant* traffic, SCOOT has the lowest ATT. SCOOTs

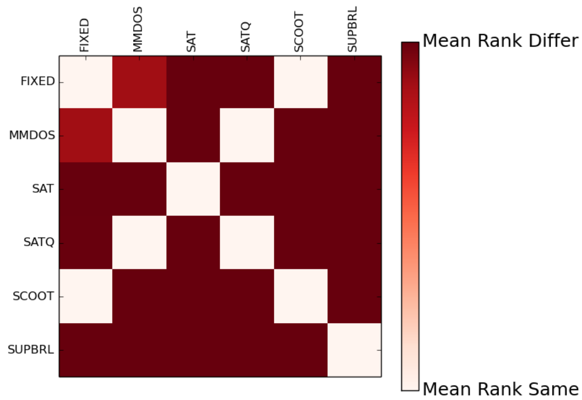
ATT is significantly different from SAT, SATQ and MMDOS in *regional* traffic but not in *constant* traffic. Lastly, SAT, which has the highest ATT in every traffic scenario, is statistically different from all the other mechanisms in unstructured traffic.



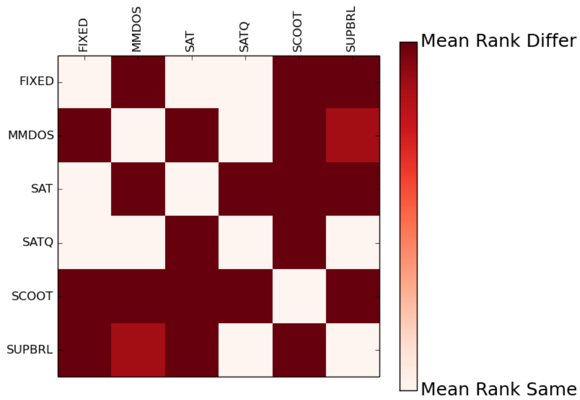
(a) Unstructured.



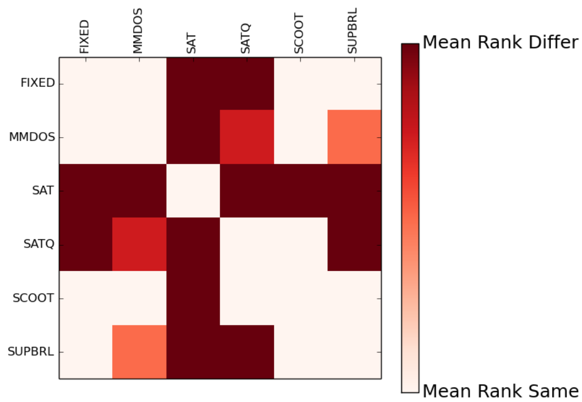
(b) Structured.



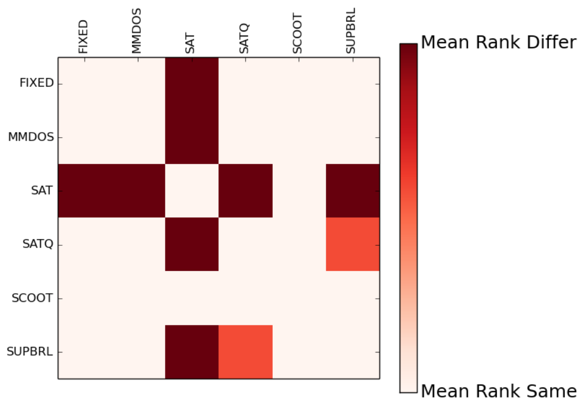
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 6.31: Visual representation of two-sample Mann-Whitney test conducted on ATT results from the 60 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

The ATT results show that the performance of SAT/Q and MMDOS in comparison to FIXED, SCOOT and SUPRL depends on the traffic scenario.

Hypothesis 1 *There will be a significant difference in* ATT *of SAT compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

—Figure 6.31 shows that SAT is statistically different from FIXED, SCOOT and SUPRL in all of the scenarios except *regional* and *constant* traffic. SAT is statistically the same as FIXED and SCOOT in *regional* and *constant* traffic, respectively (see Figure 6.31). This supports Hypothesis 1, that the ATT of SAT will be significantly different from the ATT of SCOOT, SUPRL and FIXED depending on traffic conditions.

Hypothesis 4 *There will be a significant difference in* ATT *of SATQ compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

—Figure 6.31 shows that SATQ is significantly different from FIXED and SUPRL in *unstructured*, *directional* and *structured* traffic. Figure 6.31 also shows that SATQ is significantly different from FIXED, SCOOT, and SUPRL in the *football* scenario and only significantly different from SCOOT in the *regional* traffic scenario. Lastly, in *constant* traffic, SATQ is not statistically different from FIXED, SCOOT and SUPRL.

Hypothesis 7 *There will be a significant difference in* ATT *of MMDOS compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

—Figure 6.31 shows that the ATT of MMDOS is not significantly different from the ATT of FIXED, SCOOT and SUPRL in *unstructured*, *directional*, *structured* and *constant* traffic. Also, in *football* traffic, the ATT of MMDOS is significantly different from the ATT of SCOOT and SUPRL but not the ATT of FIXED. Lastly, in *regional* traffic, the ATT of MMDOS is significantly different from the ATT of FIXED and SCOOT.

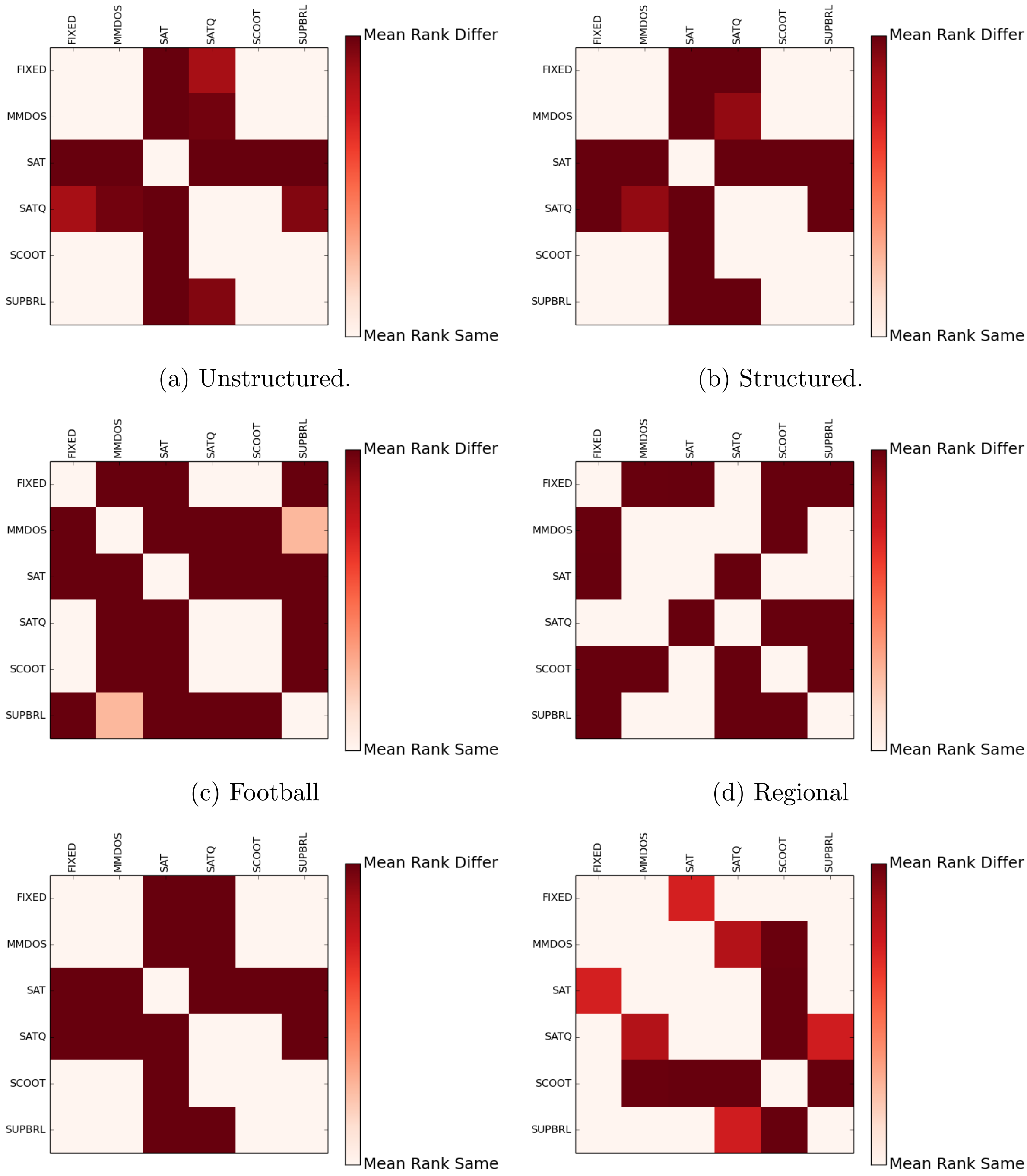
6.4.2 ATD

Average Traffic Density (ATD) (*std.*)			
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	**12.41 (0.2)**	18.34 (2.15)	21.06 (0.4)
MMDOS	13.44 (2.01)	16.65 (3.64)	21.7 (2.16)
SAT	21.21 (5.24)	16.09 (0.57)	22.1 (1.62)
SATQ	14.67 (0.98)	17.3 (0.75)	23.08 (3.19)
SCOOT	14.66 (3.64)	**14.35 (2.68)**	**20.37 (2.39)**
SUPRL	13.05 (0.92)	15.24 (2.17)	22.52 (3.11)
	Unstructured	*Football*	*Directional*
FIXED	6.61 (3.99)	8.81 (0.82)	**13.76 (0.31)**
MMDOS	5.33 (3.43)	7.78 (1.18)	14.07 (1.47)
SAT	22.7 (9.91)	13.71 (1.79)	18.52 (2.25)
SATQ	**4.47 (1.76)**	8.56 (1.11)	15.85 (0.75)
SCOOT	7.11 (5.3)	8.73 (1.3)	15.49 (3.79)
SUPRL	5.66 (2.96)	**7.55 (1.45)**	13.8 (0.98)

TABLE 6.9: Average traffic density (ATD) for each mechanism and traffic scenario.

Table 6.9 contains the ATT across both maps for the mechanisms evaluated in this chapter. The ATD results for SAT, SATQ and MMDOS mirror the ATT results. In unstructured traffic SATQ has the lowest ATD. However, Figure 6.32a shows that the ATD of SATQ is not significantly different from the ATD of FIXED, SCOOT and SUPRL. MMDOS outperforms SAT and SATQ in all of the traffic scenarios except *regional* and *unstructured* traffic. In regional MMDOS outperforms SATQ but not SAT and in *unstructured* traffic, it outperforms SAT but not SATQ. Figure 6.32 shows that MMDOS is significantly different from SAT/Q in the traffic scenarios with *unpredictable* traffic flow and not significantly different from SAT/Q in the scenarios with *predictable* traffic flow. A comparison of the market-based mechanisms to SCOOT show that in four of the traffic scenarios they outperform SCOOT, Table 6.9. In *unstructured* and *football* traffic, MMDOS and SATQ outperforms SCOOT while in *structured* and *directional* traffic, only MMDOS outperforms SCOOT.

Table 6.9 also shows that in all of the traffic scenarios, excluding *unstructured* traffic, one of the benchmarks have the lowest ATD (highlighted in bold). Additionally, in *structured* and directional traffic, FIXED has the lowest ATD. In some of traffic scenarios, the mean ATD difference is small, e.g., in *directional* traffic, the difference between the mean of FIXED (which has the lowest ATD) and SAT (which has the highest ATD) is 4.76 (vehicles per kilometre). Thus, the differences in performance in terms of ATD are slight.



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

(e) Directional

(f) Constant

FIGURE 6.32: Visual representation of two-sample Mann-Whitney test conducted on ATD results from the 60 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

The ATD results show that the performance of SAT/Q and MMDOS in comparison to FIXED, SCOOT and SUPRL depends on the traffic scenario.

Hypothesis 2 *There will be a significant difference in* ATD *of SAT compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

—In all of the traffic scenarios, except *regional* and *constant*, SAT is significantly different from FIXED, SCOOT and SUPRL. In *regional* and *constant* traffic, SAT is significantly different from FIXED and SCOOT respectively but not SUPRL.

Hypothesis 5 *There will be a significant difference in* ATD *of SATQ compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

—SATQ is significantly different from FIXED in *directional* and *structured* traffic. SATQ is significantly different from SCOOT in *regional* and *constant*. Lastly, SATQ is significantly different from SUPRL in all of the traffic scenarios except *constant*.

Hypothesis 8 *There will be a significant difference in* ATD *of MMDOS compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

—In *regional* and *football* traffic, the ATD of MMDOS is significantly different from the ATD of FIXED and SCOOT but not the ATD SUPRL. Lastly, in *constant* traffic, the ATD of MMDOS is only significantly different from the ATD of SCOOT.

6.4.3 ANS

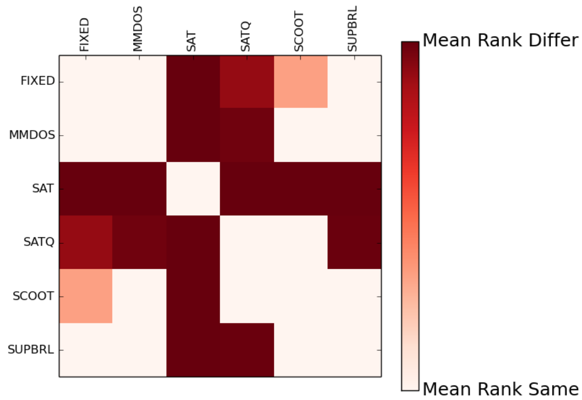
Average Number of Stops (ANS) (*std.*)			
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	**115.68 (45.88)**	189.45 (91.85)	205.33 (75.29)
MMDOS	138.23 (85.4)	**126.88 (77.59)**	**182.3 (45.18)**
SAT	359.75 (213.26)	177.73 (54.86)	245.78 (71.6)
SATQ	154 (67.08)	165.58 (52.6)	218.37 (44.99)
SCOOT	189.98 (131.28)	128.87 (75.75)	223.12 (126.29)
SUPRL	125.98 (59.9)	132.07 (69.65)	203.12 (43.92)
	Unstructured	*Football*	*Directional*
FIXED	48.58 (23.13)	96.83 (47.16)	133.03 (54.56)
MMDOS	38.18 (36.7)	80.73 (47.76)	135.52 (75.16)
SAT	392.32 (40.65)	208.23 (44.73)	284.73 (134.9)
SATQ	**31.05 (3.12)**	91.63 (45.29)	166.52 (66.6)
SCOOT	54.62 (51.38)	96.13 (58.91)	204.75 (141.49)
SUPRL	38.12 (10.22)	**76.43 (46.83)**	**131.62 (63.31)**

TABLE 6.10: Average *number of stops* (ANS) for each mechanism and traffic scenario.

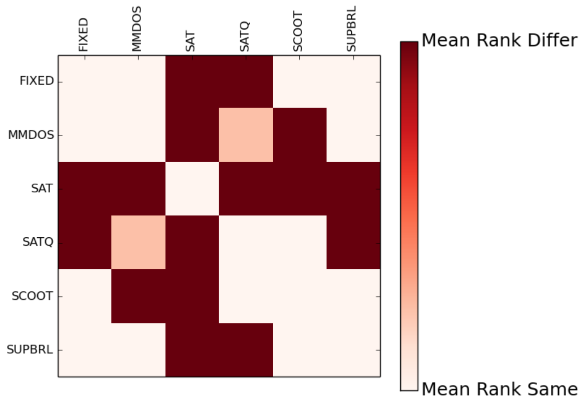
The ANS results for MMDOS and SATQ are better than their ATT and ATD results. In *unstructured* traffic SATQ has the lowest ANS and in *regional* and *constant* traffic, MMDOS has the lowest ANS.

MMDOS and SATQ also perform better in terms of ANS than ATT and ATD in comparison to SCOOT. In all of the traffic scenarios, excluding *regional* traffic, MMDOS and SATQ have lower ANS than SCOOT. In *regional* traffic, of the three market-based approaches only MMDOS has lower ANS than SCOOT. Additionally, MMDOS outperforms SAT and SATQ in all of the traffic scenarios except *unstructured* traffic.

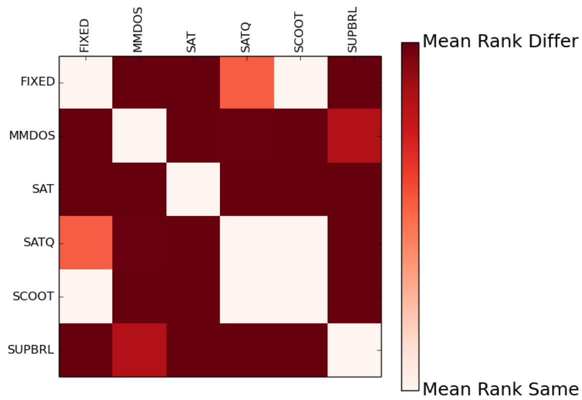
Table 6.10 does show however, that in three of the traffic scenarios, one of the benchmarks have the lowest ANS (highlighted in bold). In *football* and *directional* traffic SUPRL has the lowest ANS and in *structured* traffic, FIXED has the lowest ANS.



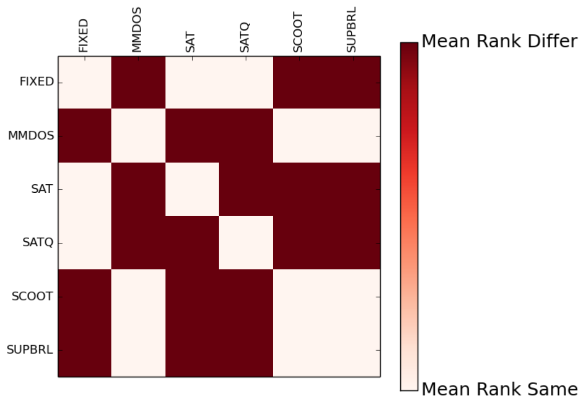
(a) Unstructured.



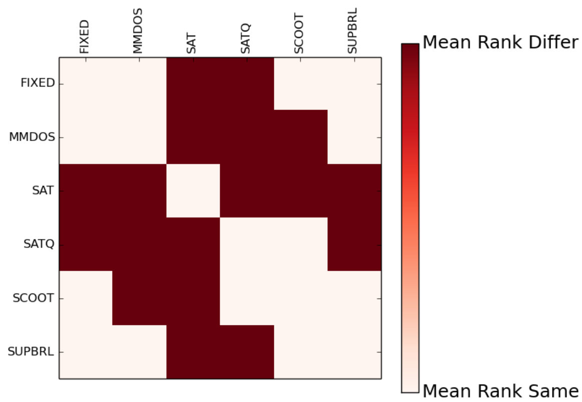
(b) Structured.



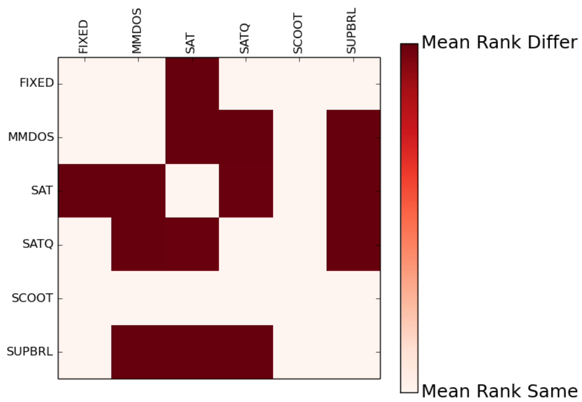
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 6.33: Visual representation of two-sample Mann-Whitney test conducted on ANS results from the 60 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

The ANS results show that the performance of SAT/Q and MMDOS in comparison to FIXED, SCOOT and SUPRL depends on the traffic scenario.

Hypothesis 3 *There will be a significant difference in* ANS *of SAT compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

—The ANS of SAT is statistically different from the ANS of FIXED, SCOOT and SUPRL in all of the traffic scenarios except *regional* and *constant* traffic. In *regional* traffic, the ANS of SAT is significantly different from SCOOT and SUPRL. Lastly, the ANS of SAT is significantly different from the ANS of FIXED and SUPRL in *constant* traffic.

Hypothesis 6 *There will be a significant difference in* ANS *of SATQ compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

—The ANS of SATQ is only significantly different from the ANS of FIXED in *directional* and *structured* traffic and significantly different from the ANS of SCOOT in *regional* traffic only. Lastly, the ANS of SATQ is statistically different from the ANS of SUPRL in every traffic scenario.

Hypothesis 9 *There will be a significant difference in* ANS *of MMDOS compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

—The ANS of MMDOS is significantly different from FIXED in *football* and *regional* traffic. The ANS of MMDOS is significantly different from the ANS of SCOOT in *football*, *directional* and *structured* traffic. Lastly, MMDOS is only significantly different from SUPRL in *constant* traffic.

Results show that the difference in performance of the mechanisms depends on traffic scenario. SATQ outperforms the other market-based mechanisms and the benchmarks in *unstructured* traffic. However, in the other traffic scenarios, MMDOS, which makes finer adjustments to all three traffic control parameters, outperforms SAT/Q. Additionally, MMDOS and SATQ outperform SCOOT in a majority of the traffic scenarios, regardless of the performance metric. Furthermore, in two of the traffic scenarios, my market-based approach has lower ANS than SUPRL (recall SUPRL is designed to reduce stops). The next chapter evaluates the performance of adjusting other combinations of *split*, *cycle* and *offset*.

Chapter 7

Results of Traffic Control Parameter Experiments

7.1 Introduction

This chapter presents the traffic simulation results for the evaluations of alternative combinations of *split*, *cycle* and *offset* usage in MMDOS and SCOOT. The MMDOS and SCOOT variants are alternative versions of MMDOS and SCOOT where the mechanisms do not adjust all three traffic control parameters, i.e, *split*, *cycle* and *offset*. The variants are named after which traffic control parameter(s) is utilised to manage traffic, e.g., MMDOS(S) only adjust the *split* while MMDOS(SC) adjust a combination of *split* and *cycle*. In this chapter, MMDOS, SCOOT and their respective variants are compared to one another as well as FIXED, SCOOT and SUPRL. FIXED does not adjust any traffic control parameters, that is, the traffic signal timing is static and SUPRL is a Reinforcement-learning based traffic controller which only adjust the *split*. The mechanisms, traffic scenarios and maps presented in this chapter are shown in Table 7.1.

This chapter is organised into two major parts, the first presents and analyses the results for each map, Phoenix (Section 7.2) and Portland (Section 7.3), and the second, is analysis across both maps (Section 7.4). Traffic performance is measured using three metrics: *average travel time*, *traffic density*, and *number of stops*. Thus, Section 7.2 presents results from traffic simulations executed on the Phoenix map and is divided into three sub-sections, one for each metric. Likewise, Section 7.3 is divided into three sub-sections, one for each metric, but for traffic simulations executed on the Portland map. Furthermore, traffic performance is evaluated using six traffic scenarios: *structured*, *unstructured*, *football*, *directional*, *constant*, and *regional*.

Mechanisms	*Traffic Scenarios*	*Maps*
MMDOS	Structured	Phoenix
MMDOS(S)	Unstructured	Portland
MMDOS(O)	Directional	
MMDOS(C)	Regional	
MMDOS(SC)	Football	
MMDOS(OC)	Constant	
MMDOS(SO)		
SCOOT		
SCOOT(S)		
SCOOT(O)		
SCOOT(C)		
SCOOT(SC)		
SCOOT(OC)		
SCOOT(SO)		
FIXED		
SUPRL		

TABLE 7.1: List of mechanisms, traffic flows and maps presented in this chapter.

Lastly, Section 7.4 contains a summary of the results and addresses the following hypothesis:

Hypothesis 10 *Adjusting alternative combinations of* split, cycle *and* offset *in MMDOS will have a significant effect on* ATT.

Hypothesis 11 *Adjusting alternative combinations of* split, cycle *and* offset *in MMDOS will have a significant effect on* ATD.

Hypothesis 12 *Adjusting alternative combinations of* split, cycle *and* offset *in MMDOS will have a significant effect on* ANS.

Hypothesis 13 *Adjusting alternative combinations of* split, cycle *and* offset *in SCOOT will have a significant effect on* ATT.

Hypothesis 14 *Adjusting alternative combinations of* split, cycle *and* offset *in SCOOT will have a significant effect on* ATD.

Hypothesis 15 *Adjusting alternative combinations of* split, cycle *and* offset *in SCOOT will have a significant effect on* ANS.

7.2 Results: Phoenix

This section presents the results of the experiments executed on the Phoenix map. This section is divided into sub-sections, covering each of the three traffic performance metrics: *average travel time* (Section 7.2.1), *traffic density* (Section 7.2.3), and *number of* *stops* (Section 7.2.6). The traffic control systems are evaluated in three traffic scenarios with *predictable* traffic flow (*structured*, *regional*, and *constant*) and three traffic scenarios with *unpredictable* traffic flow (*unstructured*, *football*, and *directional*). *Unstructured*, *football*, and *directional* are *unpredictable* in that they represent traffic conditions where high levels of traffic demand is not along a single artery and in a single direction. The Mann-Whitney test is used to determine statistical significance between traffic performance results. The threshold value of $p = .05$ was used to determine whether the null hypotheses (the samples were the same) was rejected. The Mann-Whitney test results are presented in a visual manner in lieu of tables to provide the same information but in a more compact manner than a large table(s).

7.2.1 Travel Time (ATT)

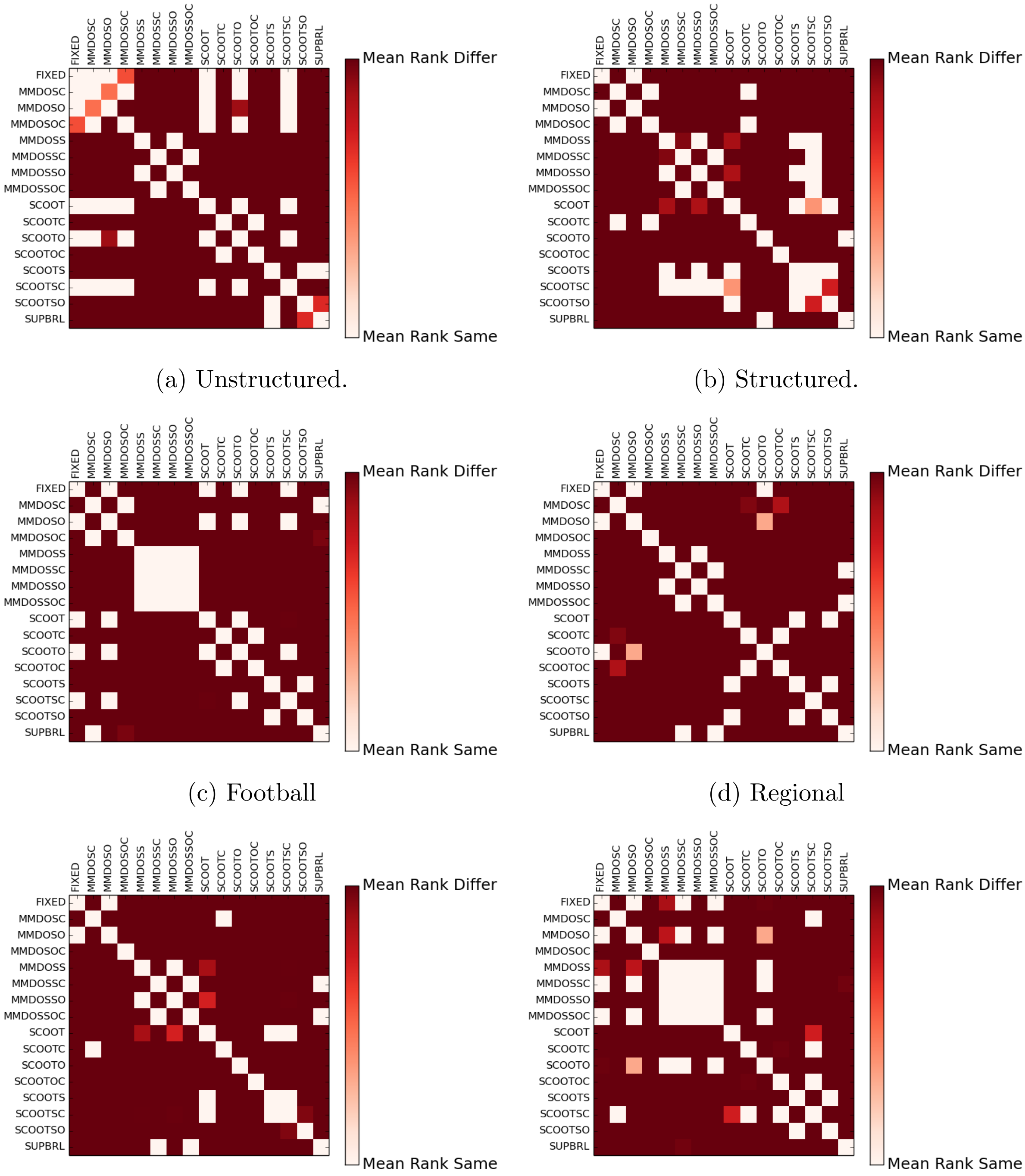
Table 7.2 shows that MMDOS(SO) has the lowest ATT in *unstructured* traffic of all the mechanisms. Figure 7.1a shows that MMDOS(SO) performance is significantly different from all the other mechanisms, except for MMDOS(S) which has the second lowest ATT in *unstructured* traffic. MMDOS(SO) and MMDOS(S) outperforms SCOOT, SUPRL and FIXED in *unstructured* traffic on the Phoenix map. In the SCOOT group, SCOOT(SO) has the lowest ATT; it is significantly lower than FIXED but not SUPRL. Also, SCOOT(SO) does not have lower ATT than MMDOS(SO) and MMDOS(S). In the *football* scenario, MMDOS(C) and SUPRL have similar ATT results, Figure 7.1b, although, SURPL has lower ATT than MMDOS(C). However, in the *football* scenario, SCOOT(S) has the overall lowest ATT. In *directional* traffic, MMDOS(OC) has the lowest ATT amongst all the mechanisms and the results are significant, see Figure 7.1e. SCOOT(OC) which outperforms both SUPRL and FIXED, has the lowest ATT amongst the SCOOT variants in *directional* traffic.

Table 7.2 also shows that the MMDOS and SCOOT variants perform well in the scenarios with predictable traffic. In *structured* traffic, MMDOS(C) has the lowest ATT amongst the MMDOS variants, however, Figure 7.1b shows that it is not significantly different from MMDOS(OC) and SCOOT(C). SCOOT(OC) has the over all lowest ATT in *structured* traffic.

Amongst the MMDOS variants, in *regional* and *constant* traffic, MMDOS(C) has the lowest ATT. In both scenarios, MMDOS(C) outperforms SUPRL and FIXED, although, not SCOOT in the *regional* traffic scenario. SCOOT(SC) and SCOOT(OC) has the lowest ATT within the SCOOT variants in *regional* and *constant*, respectively.

Average Travel Time (ATT) (*std.*)			
		Traffic Pattern	
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	166.11 (1.09)	184.07 (1.32)	184.6 (0.2)
MMDOS(C)	**128.44 (0.39)**	136.45 (0.62)	138.27 (0.17)
MMDOS(O)	166.02 (1.19)	183.59 (1.74)	184.73 (0.38)
MMDOS(OC)	128.46 (0.32)	134.02 (0.35)	**134.46 (0.22)**
MMDOS(S)	147.62 (0.63)	151.11 (5.38)	187.8 (5.47)
MMDOS(SC)	149.86 (2.85)	144.18 (0.5)	193.64 (16.17)
MMDOS(SO)	147.46 (0.76)	149.66 (4.58)	189.86 (5.32)
MMDOS	150.27 (2.72)	144.28 (0.51)	190.83 (11.86)
SCOOT	144.8 (3.44)	129.42 (3.71)	144.7 (3.52)
SCOOT(C)	128.58 (0.45)	136.98 (0.48)	139.93 (0.25)
SCOOT(O)	160.17 (1.59)	184.82 (1.25)	185.3 (0.93)
SCOOT(OC)	126.36 (1)	136.92 (0.46)	139.65 (0.31)
SCOOT(S)	146.22 (3.53)	**127.93 (1.66)**	173.24 (9.23)
SCOOT(SC)	148.57 (5.41)	125.04 (3.37)	141.57 (6.18)
SCOOT(SO)	144.56 (3.37)	128.42 (1.77)	175.15 (12.81)
SUPRL	159.48 (1.3)	144.03 (1.42)	206.12 (7.85)
	Unstructured	*Football*	*Directional*
FIXED	1108.81 (168.99)	190.89 (12.8)	173.18 (0.98)
MMDOS(C)	1257.48 (283.07)	143.29 (6.2)	131.49 (0.42)
MMDOS(O)	1054.47 (150.27)	192.05 (14.5)	172.97 (0.97)
MMDOS(OC)	1198.94 (94.36)	147.02 (4.35)	**129.95 (0.25)**
MMDOS(S)	518.98 (16.81)	156.8 (4.17)	149.28 (0.67)
MMDOS(SC)	708.08 (69.72)	154.38 (4.46)	160.79 (3.33)
MMDOS(SO)	**515.51 (17.43)**	157.07 (4.62)	149.13 (0.69)
MMDOS	717.92 (79.4)	154.28 (4.87)	160.17 (3.49)
SCOOT	1231.36 (369.63)	184.81 (7.66)	146.93 (5.16)
SCOOT(C)	1719.6 (145.43)	229.46 (10.67)	131.6 (0.52)
SCOOT(O)	1257.28 (260.09)	192.51 (12.52)	169.83 (1.4)
SCOOT(OC)	1660.44 (296.11)	230.3 (13.41)	131.02 (0.38)
SCOOT(S)	839.91 (69.53)	133.44 (3.46)	146.03 (3.35)
SCOOT(SC)	1188.66 (199.5)	193.72 (8.36)	145.59 (4.24)
SCOOT(SO)	790.36 (62.82)	**133.81 (2.7)**	141.9 (3.23)
SUPRL	855.66 (78.43)	142.76 (4.05)	160.4 (1.26)

TABLE 7.2: Average travel times (ATT) for each mechanism and traffic scenario.



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

(e) Directional

(f) Constant

FIGURE 7.1: Visual representation of two-sample Mann-Whitney test conducted on ATT (Phoenix map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

7.2.2 Cumulative Average Travel Time (CATT)

In *unstructured* traffic, MMDOS(S) and MMDOS(SO) have the lowest CATT throughout the entire scenario, see Figure 7.2a. MMDOS(S) and MMDOS(SO) display little change in CATT during the *unstructured* traffic disruption. Also, prior to the disruption in *unstructured* traffic, MMDOS and MMDOS(SC) performs similar to SCOOT and SUPRL. Although both MMDOS and MMDOS(SC) have an increase in CATT during the disruption, both mechanisms maintain lower CATT than SCOOT, SUPRL and FIXED for the remainder of the scenario. Additionally, on the Phoenix map, in the *unstructured* traffic scenario, MMDOS(OC) and MMDOS(C) have the highest CATT.

In the *football* scenario, MMDOS(OC) and MMDOS(C) have the lowest CATT. Although SUPRL has slightly higher CATT than MMDOS(OC) and MMDOS(C), during the second disruption and onwards, its CATT is similar to MMDOS(OC) and MMDOS(C). Additionally, all the MMDOS variants have lower CATT than SCOOT in the *football* scenario. MMDOS(C) and MMDOS(OC) have the lower CATT than the other mechanisms in *directional* traffic, see Figure 7.2e. Additionally, in *directional* traffic on the Phoenix map, MMDOS, MMDOS(SC), MMDOS(SO) and MMDOS(S) have lower CATT than SUPRL but not lower than SCOOT.

On the Phoenix map, MMDOS(C) and MMDOS(OC) have the lowest CATT in *structured* and *constant* traffic. In *structured* traffic, on the Phoenix map, Figure 7.2b, MMDOS(C) and MMDOS(OC) are the only MMDOS variants that have lower CATT than SCOOT. On the Phoenix map, in *structured* traffic MMDOS(O) which performs similar to FIXED has greater CATT during and after the *structured* traffic disruption, see Figure 7.2b. Also, in *constant* traffic all the variants have lower CATT than SUPRL, see Figure 7.2f. In the *regional* traffic scenario, SCOOT has the lowest CATT, see Figure 7.2d. Also, in *regional* traffic, MMDOS(C) and MMDOS(OC) have lower CATT than SUPRL.

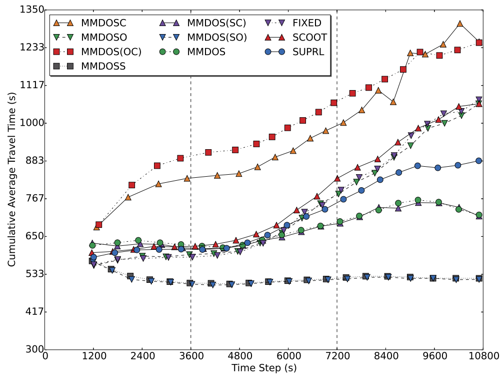
In *unstructured* traffic, SCOOT(SO), SCOOT(S) and FIXED have the lowest CATT prior to the disruption, see Figure 7.3a. During the *unstructured* traffic disruption all the mechanisms have an increase in CATT, however, SCOOT(SO) and SCOOT(S) maintain lower CATT than all the other mechanisms even after the disruption ends. After the disruption terminates, SCOOT(SO) maintains lower CATT than all of the other mechanisms. Lastly, SCOOT(C) and SCOOT(OC) have the highest CATT in *unstructured* traffic.

SCOOT(SO) and SCOOT(S) also perform well in the *football* scenario, see Figure 7.3c. In the *football* scenario, SCOOT(SO) and SCOOT(S) have the lowest CATT throughout the entire scenario, showing little change in CATT even during the disruptions. Also, in the *football* scenario, SCOOT(O) which performs similar to FIXED, has lower CATT than SCOOT but SUPRL. Lastly, in the *football*, SCOOT(OC) and SCOOT(C) have the highest CATT.

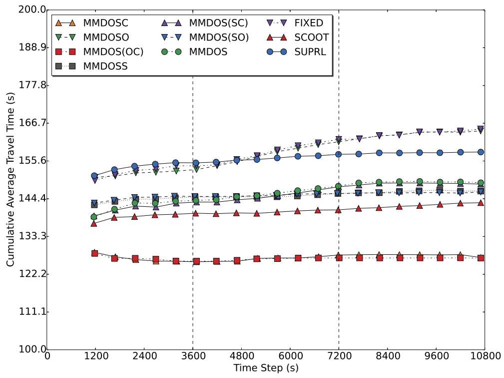
However, in *directional* traffic, SCOOT(OC) and SCOOT(C) have the lowest CATT, see Figure 7.3e. In *directional* traffic, on the Phoenix map, all the SCOOT variants (excluding SCOOT(OC) and SCOOT(C)) have higher CATT than SCOOT but less than FIXED. Figure 7.3e also shows that only SCOOT(O) is the only mechanism with CATT greater than SUPRL.

SCOOT(OC) and SCOOT(C) also have the lowest CATT in *structured* traffic, see Figure 7.3b. In *structured* traffic, all the SCOOT variants perform as well as or better than SUPRL, however, only SCOOT(OC) and SCOOT(C) have CATT lower than SCOOT. In *regional* traffic, SCOOT(SC) has the lowest CATT, displaying little change during the disruption. SCOOT(OC) and SCOOT(C) has lower CATT than SUPRL but not SCOOT in *regional* traffic.

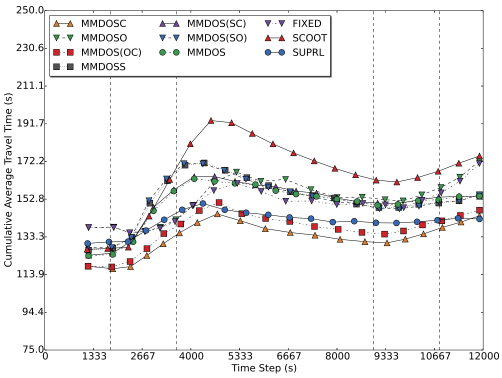
On the Phoenix map, in *regional* traffic, the CATT of SCOOT(SO) and SCOOT(S) is similar to the CATT of SCOOT. Lastly, in *regional* traffic, SCOOT(O) performs similar to FIXED, the two have the highest CATT in *regional* traffic on the Phoenix map. In *constant* traffic, SCOOT(OC) and SCOOT(C) has the lowest CATT as well, however, the CATT of SCOOT and SCOOT(SC) is only marginally higher, see Figure 7.3f. Additionally, all the SCOOT variants have lower CATT than SUPRL Finally, SCOOT(O) also performs similar to FIXED in *constant* traffic but both mechanisms have lower CATT than SUPRL.



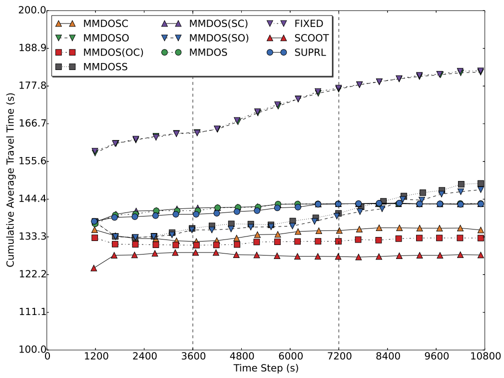
(a) Unstructured.



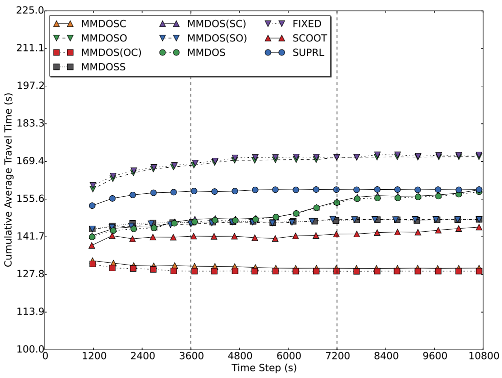
(b) Structured.



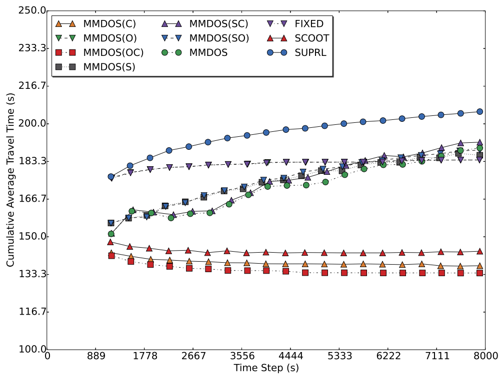
(c) Football



(d) Regional

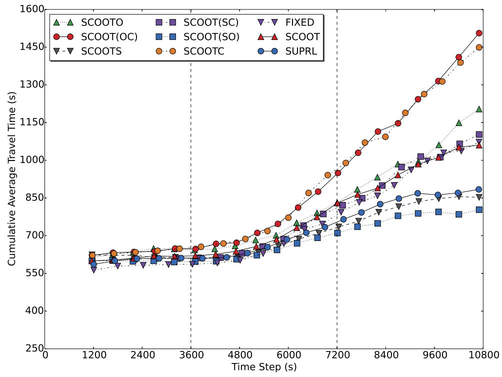


(e) Directional

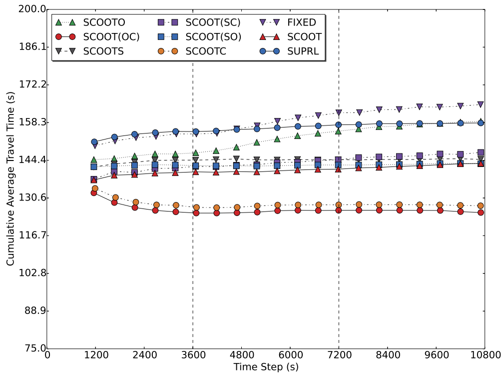


(f) Constant

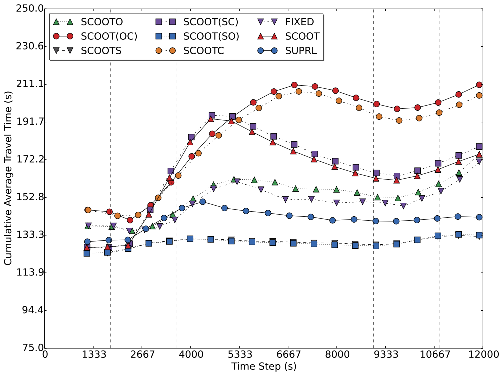
FIGURE 7.2: Cumulative average travel times (over 30 simulations) on the Phoenix map. Beginning and ending of disruptions are marked by dotted lines.



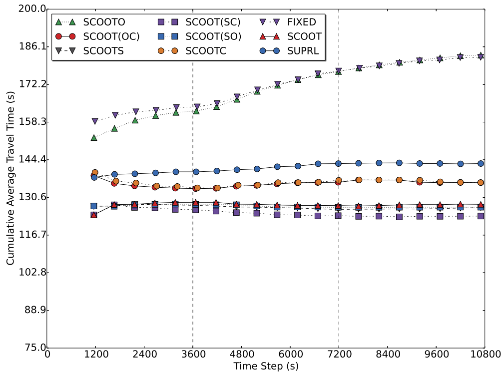
(a) Unstructured.



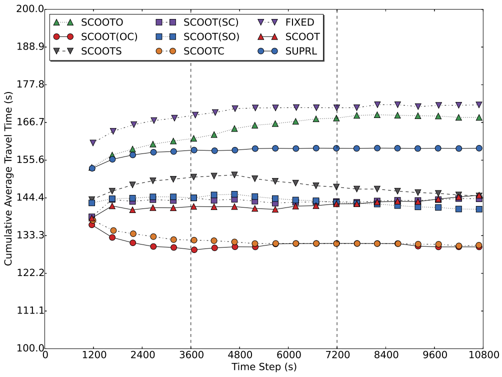
(b) Structured.



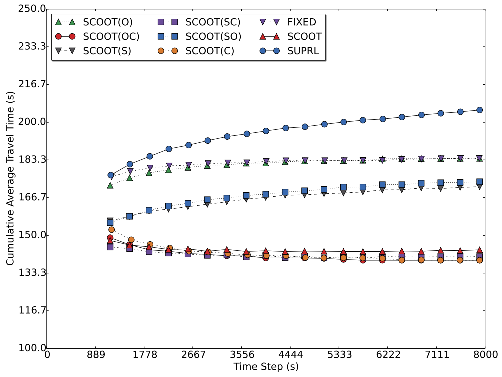
(c) Football



(d) Regional



(e) Directional



(f) Constant

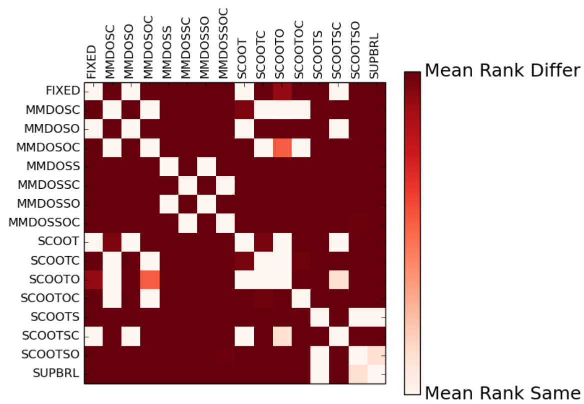
FIGURE 7.3: Cumulative average travel times (over 30 simulations) on the Phoenix map. Beginning and ending of disruptions are marked by dotted lines.

7.2.3 Density (ATD)

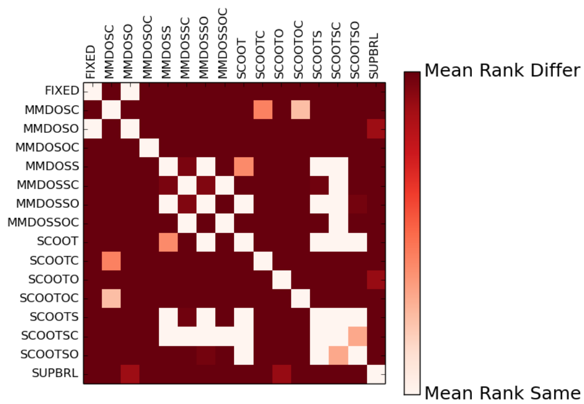
On the Phoenix map with *unstructured* traffic, several of the MMDOS have very low ATD (below 10 vehicles per kilometre, *vpk*). Although not significantly different from FIXED, Table 7.3 shows that MMDOS(SO) has the over all lowest ATD. SCOOT(S) has the lowest ATD amongst the SCOOT variants in *unstructured* traffic. In *football*, *directional* and *structured* traffic, there is little difference between the ATD performance of the mechanisms. For example, in *football* traffic, the range of ATD is only 3 *vpk*. Nonetheless, SCOOT(SO) has the lowest ATD of all the mechanisms in *football* traffic while MMDOS(C) has the lowest ATD amongst the MMDOS variants in *football* traffic. In *directional* traffic, MMDOS(OC) has the lowest ATD of all the mechanisms and the results are significant, see Figure 7.4e. SCOOT(OC) has the lowest ATD amongst the SCOOT variants in *directional* traffic. MMDOS(OC) also has the overall lowest ATD in *structured* and *constant* traffic, Table 7.3. In both scenarios MMDOS(OC) performance is significantly different from all the other mechanisms, Figures 7.4b and 7.4f. The *offset* and *cycle* combination of parameters also worked well for the SCOOT variants in *structured* and *constant* traffic. SCOOT(OC) has the lowest ATD amongst the SCOOT variants in *structured* and *constant* traffic. Lastly, in *regional* traffic, MMDOS(OC) and SCOOT(SC) have the lowest ATD in their respective groups. In *structured*, *regional* and *constant* traffic, the MMDOS and SCOOT variants have lower ATD than FIXED, SCOOT and SUPRL.

Average Traffic Density (ATD) (*std.*)			
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	12.39 (0.19)	16.22 (0.25)	21.44 (0.09)
MMDOS(C)	9.86 (0.11)	12.51 (0.15)	16.77 (0.06)
MMDOS(O)	12.34 (0.2)	16.12 (0.34)	21.45 (0.11)
MMDOS(OC)	**9.51 (0.12)**	11.78 (0.12)	**15.7 (0.09)**
MMDOS(S)	11.24 (0.09)	13.39 (0.57)	23.43 (0.72)
MMDOS(SC)	11.42 (0.25)	13 (0.15)	23.84 (1.69)
MMDOS(SO)	11.22 (0.13)	13.34 (0.46)	23.72 (0.74)
MMDOS	11.49 (0.24)	13.05 (0.14)	23.6 (1.27)
SCOOT	11.07 (0.26)	11.72 (0.38)	18.07 (0.44)
SCOOT(C)	9.94 (0.11)	12.67 (0.13)	17.28 (0.08)
SCOOT(O)	11.98 (0.21)	16.27 (0.22)	21.6 (0.14)
SCOOT(OC)	9.77 (0.12)	12.61 (0.11)	17.25 (0.08)
SCOOT(S)	11.14 (0.3)	11.56 (0.2)	21.58 (1.19)
SCOOT(SC)	11.34 (0.46)	**11.34 (0.33)**	17.68 (0.77)
SCOOT(SO)	11.01 (0.3)	11.58 (0.22)	21.84 (1.56)
SUPRL	12.15 (0.17)	13.1 (0.17)	25.51 (1.01)
	Unstructured	*Football*	*Directional*
FIXED	10.46 (1.31)	8.08 (0.51)	13.5 (0.15)
MMDOS(C)	17.38 (7.41)	6.25 (0.29)	10.44 (0.12)
MMDOS(O)	10.08 (1.28)	8.15 (0.58)	13.47 (0.16)
MMDOS(OC)	19.65 (8.34)	6.41 (0.22)	**10.1 (0.12)**
MMDOS(S)	5.24 (0.28)	6.76 (0.22)	11.96 (0.13)
MMDOS(SC)	7.08 (0.71)	6.64 (0.23)	12.79 (0.29)
MMDOS(SO)	**5.19 (0.26)**	6.77 (0.25)	11.93 (0.11)
MMDOS	7.82 (3.33)	6.64 (0.24)	12.72 (0.35)
SCOOT	11.8 (3.42)	7.51 (0.33)	11.75 (0.39)
SCOOT(C)	13.68 (1.16)	9.14 (0.45)	10.54 (0.13)
SCOOT(O)	13.08 (4.16)	8.16 (0.51)	13.18 (0.15)
SCOOT(OC)	16.41 (3.57)	9.4 (0.53)	10.51 (0.12)
SCOOT(S)	8.35 (0.71)	**5.73 (0.18)**	11.69 (0.32)
SCOOT(SC)	10.79 (1.27)	7.82 (0.39)	11.71 (0.37)
SCOOT(SO)	7.95 (0.6)	5.76 (0.14)	11.38 (0.29)
SUPRL	8.53 (0.86)	6.13 (0.21)	12.85 (0.2)

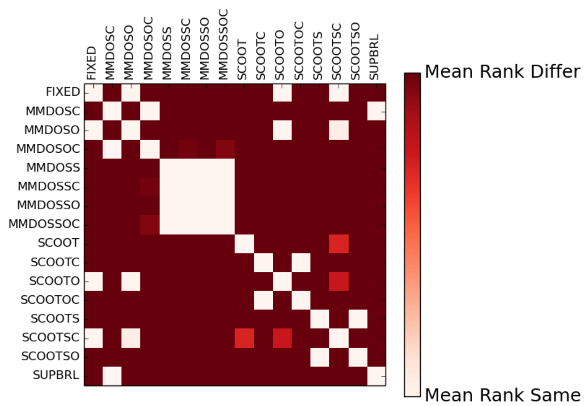
TABLE 7.3: Average traffic density (ATD) for each mechanism and traffic scenario.



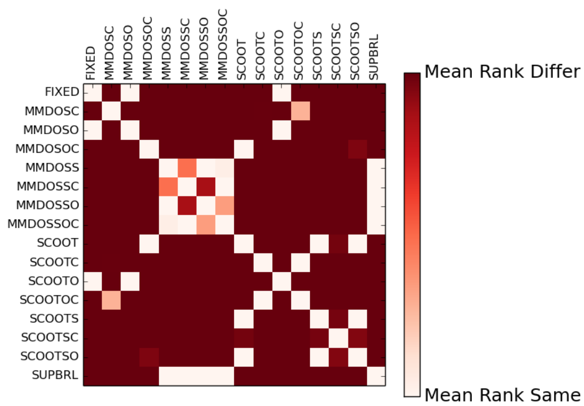
(a) Unstructured.



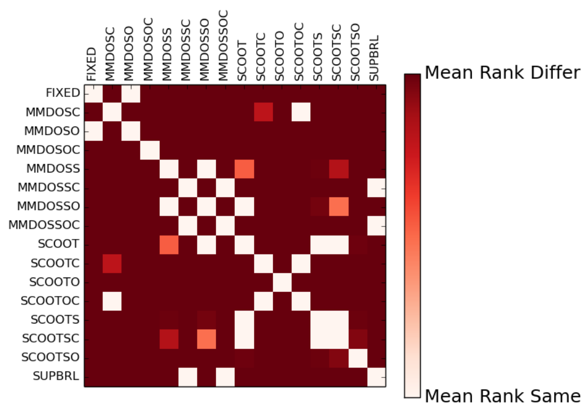
(b) Structured.



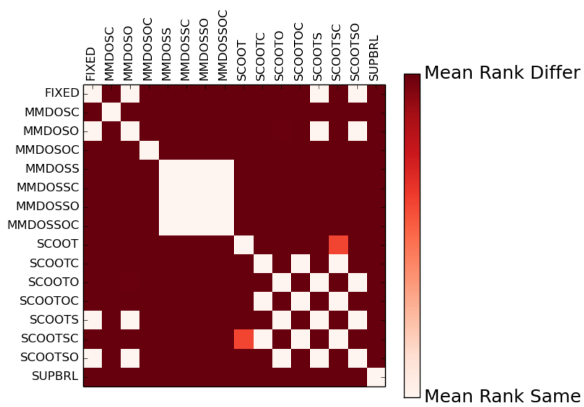
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 7.4: Visual representation of two-sample Mann-Whitney test conducted on ATD (Phoenix map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

7.2.4 Cumulative Average Density (CAD)

In *unstructured* traffic, MMDOS(SO) and MMDOS(S) have the lowest CAD throughout the entire scenario, see Figure 7.5a. Although all the mechanisms have an increase in CAD once the disruption begins, the CAD of MMDOS(SO) and MMDOS(S) remains lower than the other mechanisms. Also, in the *unstructured* traffic scenario, MMDOS and MMDOS(SC) have lower CAD than SCOOT and SUPRL during and after the disruption. Lastly, in the both maps, MMDOS(OC) and MMDOS(C) have the highest CAD but on the Phoenix map neither ends in gridlock.

In the *football* scenario, Figure 7.5c, all the variants have similar CAD during the match. However, during the disruptions there are differences in CAD. In the first disruption, all the MMDOS variants have lower CAD than SCOOT; MMDOS(C) and MMDOS(OC) have similar CAD levels to SUPRL. In the second disruption, all the MMDOS variants, excluding MMDOS(O), have lower CAD than SCOOT.

In *directional* traffic, MMDOS(OC) and MMDOS(C) have the lowest CAD on both maps, see Figure 7.5e. However, on the Phoenix map, in *directional* traffic, all the other MMDOS variants perform as well as or worse than SCOOT in terms of CAD. Additionally, on the Phoenix map, MMDOS(O) which has similar CAD to FIXED, has the highest CAD in *directional* traffic. The CAD results in *structured* traffic, Figure 7.5b, mirror the CAD results for *directional* traffic. In *structured* traffic, MMDOS(OC) and MMDOS(C) have lower CAD than the benchmarks and MMDOS(O) has the highest CAD. MMDOS(OC) and MMDOS(C) also have the lowest CAD in the *constant* traffic scenario, see Figure 7.5f. Also, in the *constant* traffic scenario, all the variants have higher CAD than SCOOT but in general, lower CAD than SUPRL. In *regional* traffic, SCOOT has the lowest CAD but periodically, MMDOS(OC) has lower CAD than SCOOT. Also, in *regional* traffic, MMDOS and MMDOS(SC) has a similar performance to SUPRL in terms of CAD. Lastly, in *regional* traffic, MMDOS(O) has the highest CAD amongst the MMDOS variants.

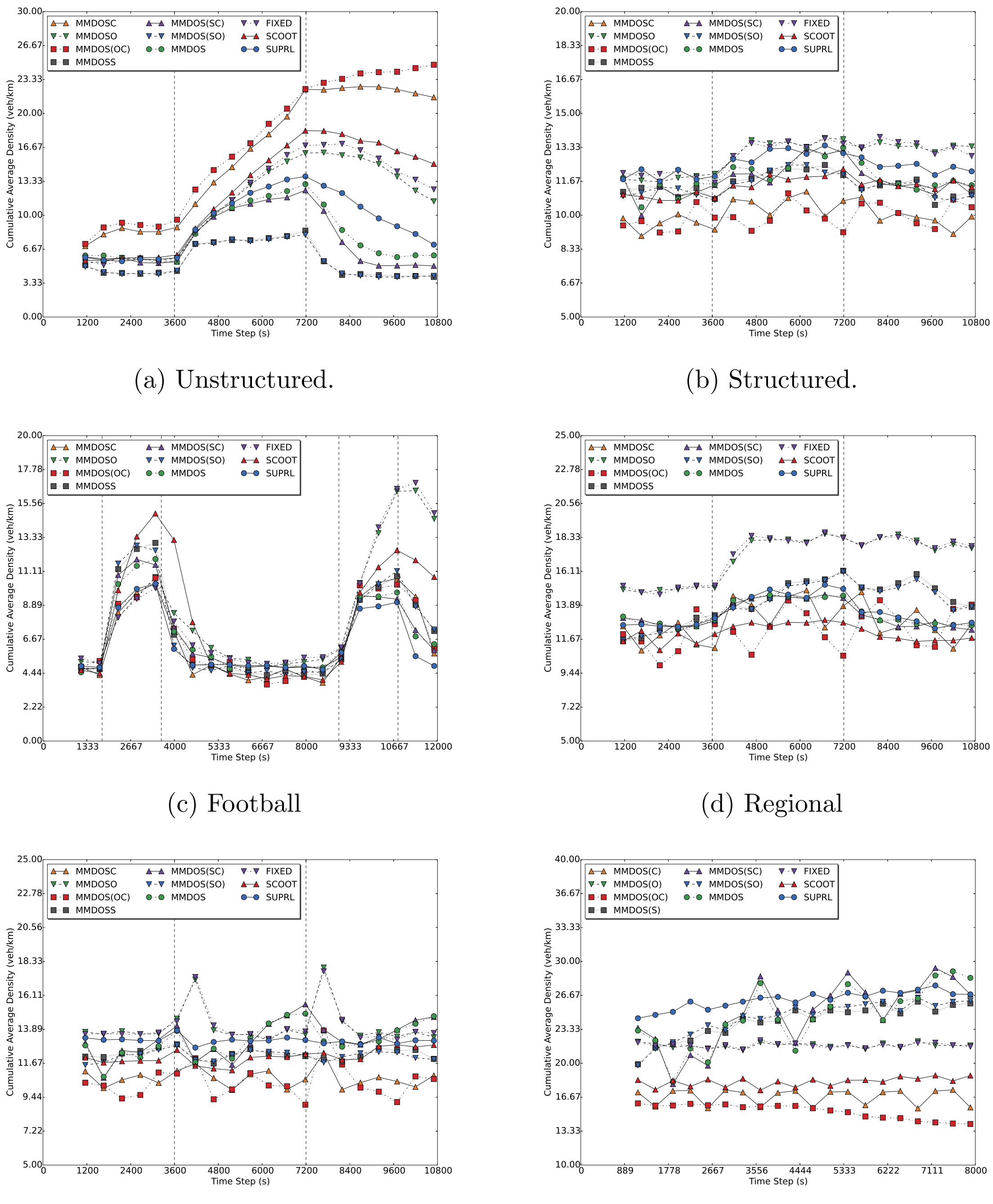
In *unstructured* traffic, prior to the disruption all the mechanisms have similar CAD, see Figure 7.6a. Although all the mechanisms have an increase in CAD once the disruption begins, SCOOT(S) and SCOOT(SO) has a lower CAD than FIXED, SCOOT and SUPRL. After the disruption, SCOOT(SO) maintains a CAD lower than the benchmarks while SCOOT(S) performs similar to SUPRL. On the Phoenix map, SCOOT(SC) has similar CAD to FIXED, however, both have lower CAD during and after the disruption. Lastly, in *unstructured* traffic, SCOOT(C) and SCOOT(OC) have higher CAD than all the other SCOOT variants.

SCOOT(S) and SCOOT(SO) also have low CAD during the disruptions in the *football* scenario, see Figure 7.6c. In the first disruption, SCOOT(S) and SCOOT(SO) have the lowest CAD, however, in the CAD of SUPRL is as low as the CAD of SCOOT(S) and SCOOT(SO). The other SCOOT variants perform as well as or better than SCOOT during the first disruption and worse than SCOOT during the second disruption. During the football match, SCOOT(S) and SCOOT(SO) reach their lowest CAD quicker than the other mechanisms. Lastly, SCOOT(OC) and SCOOT(C) have the highest CAD during the football match; both mechanisms did not reach their pre disruption levels of CAD until the disruption nearly ends.

In *directional* traffic, SCOOT(OC) and SCOOT(C) have the lowest CAD, see Figure 7.6e. All of the SCOOT variants, excluding SCOOT(O) have lower CAD than SUPRL. SCOOT also has lower CAD than SUPRL and performs similar to its variants. In *directional* traffic, SCOOT(O) has the highest CAD of all the SCOOT variants, performing similar to FIXED.

In *structured* and *constant* traffic, SCOOT(OC) and SCOOT(C) have the lowest CAD, see Figure 7.6. In *structured* traffic, SCOOT(S), SCOOT(SO), SCOOT(SC) and SCOOT have similar CAD performance. All four mechanisms have lower CAD than SUPRL, FIXED and SCOOT(O), see Figure 7.6b. In *constant* traffic, SCOOT(SC) and SCOOT have similar CAD which is near the CAD of SCOOT(OC) and SCOOT(C). Additionally, in *constant* traffic, SCOOT(O), SCOOT(SO) and SCOOT(S) perform similar to FIXED, however, all three have lower CAD than SUPRL.

Finally, in *regional* traffic, all the SCOOT variants, excluding SCOOT(O), perform similar to SCOOT, see Figure 7.6d. Although periodically SCOOT(OC) and SCOOT(C) have higher CAD than SUPRL, all the SCOOT variants, excluding SCOOT(O) have lower CAD than SUPRL in *constant* traffic on the Phoenix map.



(a) Unstructured.

(b) Structured.

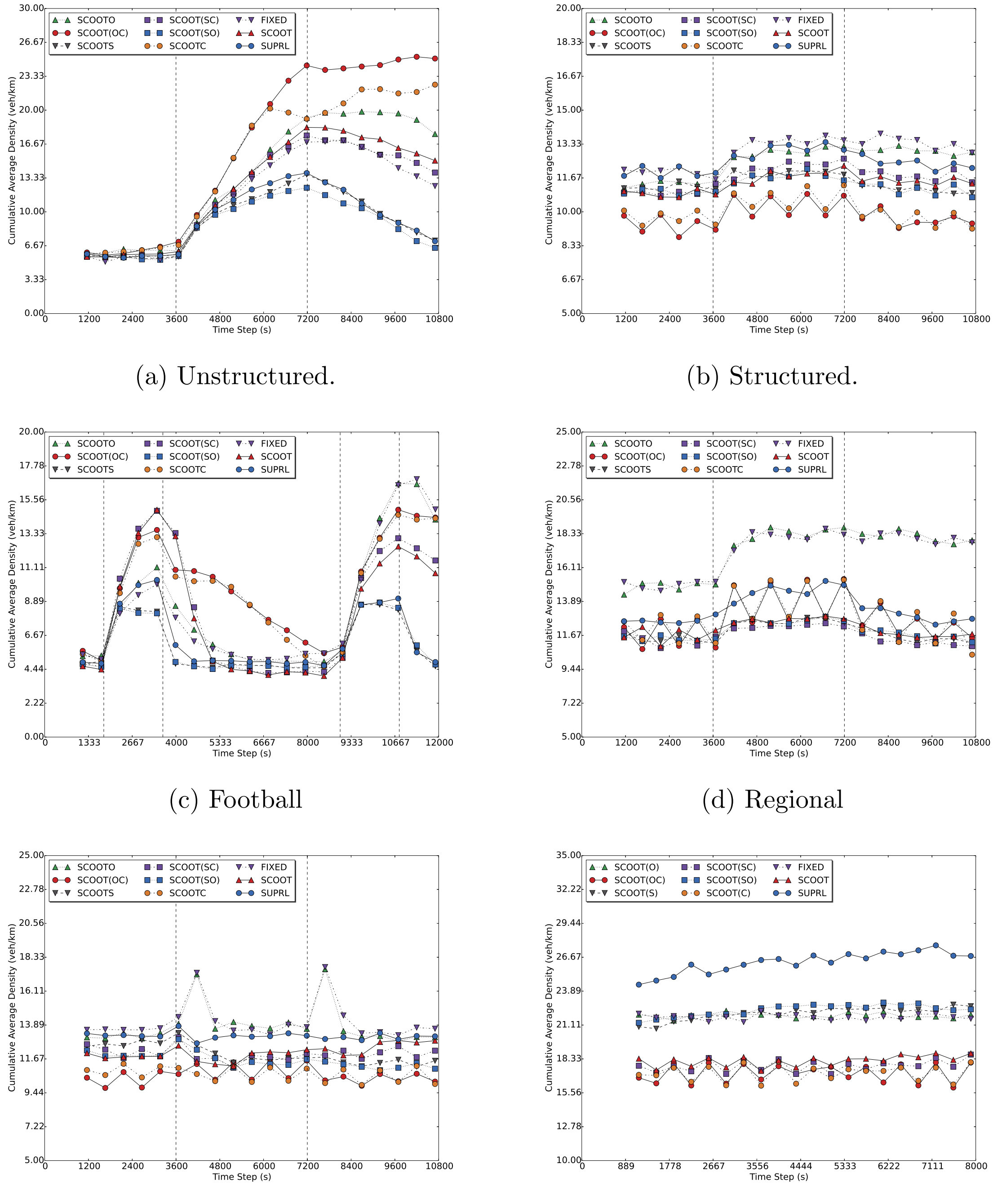
(c) Football

(d) Regional

(e) Directional

(f) Constant

FIGURE 7.5: Cumulative average density (over 30 simulations) on the Phoenix map. Beginning and ending of disruptions are marked by dotted lines.



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

(e) Directional

(f) Constant

FIGURE 7.6: Cumulative average density (over 30 simulations) on the Phoenix map. Beginning and ending of disruptions are marked by dotted lines.

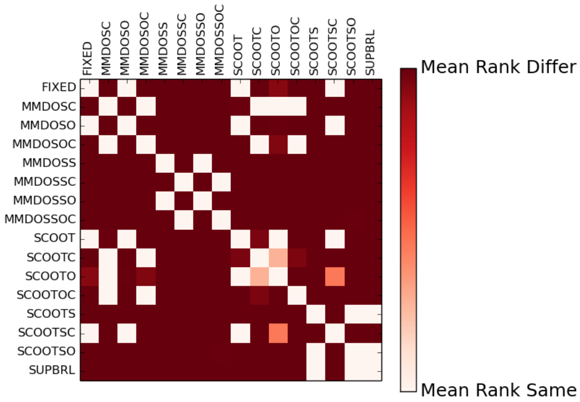
7.2.5 Vehicle Stops (ANS)

In *unstructured* traffic, MMDOS(SO) has the lowest ANS of all the mechanisms, less than half of the ANS of SUPRL. MMDOS(SO) ANS performance in *unstructured* traffic is significantly different from all the other mechanisms except MMDOS(S), see Figure 7.7a. On the Phoenix map, Table 7.4 shows that three of the MMDOS variants outperform SUPRL in terms of ANS in *unstructured* traffic. SCOOT(S) which has the lowest ANS within the SCOOT variants, did not perform significantly different from SUPRL in *unstructured* traffic. MMDOS and MMDOS(OC) have the lowest ANS amongst the MMDOS variants in *football* and *directional* traffic, respectively. However, the SCOOT variants have the lowest overall ANS in *football* and *directional* traffic. In *football* traffic, SCOOT(S) and SCOOT(OC) have the lowest ANS in *football* and *directional* traffic, respectively. However, in *directional* traffic, the difference between SCOOT(OC) and MMDOS(OC) is not significant, see Figure 7.7e.

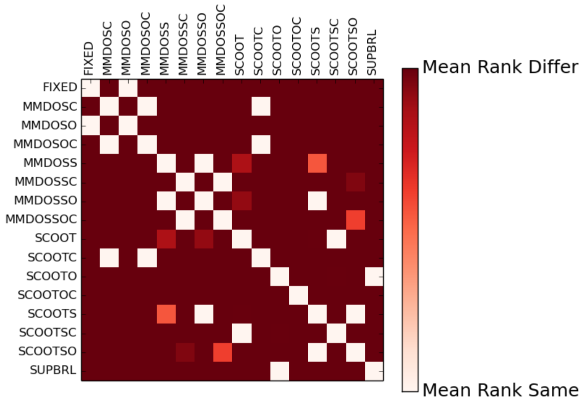
In *structured* and *constant* traffic, MMDOS(OC) has the lowest ANS amongst the MMDOS variants. In both scenarios, MMDOS(OC) outperforms SCOOT and SUPRL; the differences between the MMDOS variants and the benchmarks are significant, Figures 7.7b and 7.7f. Amongst the SCOOT variants, SCOOT(OC) has the lowest ANS in *structured* and *constant* traffic. As with ATT, in *structured* and *constant* traffic, there is an MMDOS variant that outperforms SCOOT. Lastly, in *regional* traffic, MMDOS(SOC) and SCOOT(SO) have the lowest ANS in their respective groups.

Average Number of Stops (ANS) (*std.*)			
		Traffic Pattern	
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	70.23 (1.33)	98.4 (1.92)	130.7 (0.6)
MMDOS(C)	49.77 (0.68)	67.63 (1.27)	91.57 (0.5)
MMDOS(O)	69.93 (1.46)	97.77 (2.45)	130.9 (0.66)
MMDOS(OC)	49.67 (0.76)	64.83 (0.79)	**86.4 (0.56)**
MMDOS(S)	58.47 (0.63)	74.97 (5.87)	149.63 (7.74)
MMDOS(SC)	53.43 (1.65)	**49.77 (0.9)**	140.67 (14.11)
MMDOS(SO)	58.3 (0.84)	73.97 (4.82)	152.77 (7.93)
MMDOS	53.87 (1.57)	49.97 (0.76)	138.57 (11.27)
SCOOT	60.17 (2.18)	53.97 (2.92)	98.63 (3.6)
SCOOT(C)	49.47 (0.78)	67.73 (1.01)	94.33 (0.61)
SCOOT(O)	66.03 (1.71)	98.57 (1.7)	131.47 (1.04)
SCOOT(OC)	**47.5 (1.11)**	67.4 (0.77)	93.6 (0.77)
SCOOT(S)	57.13 (3.01)	50.7 (1.97)	126.4 (12.55)
SCOOT(SC)	62.17 (4.42)	51.87 (2.75)	96 (5.92)
SCOOT(SO)	55.8 (3.24)	50.63 (1.96)	129.2 (16.83)
SUPRL	66.63 (1.4)	63.03 (1.33)	160.23 (10.39)
	Unstructured	*Football*	*Directional*
FIXED	68.33 (16.6)	50.27 (5.01)	78.97 (1.13)
MMDOS(C)	180.97 (119.46)	34.07 (2.99)	54.27 (0.91)
MMDOS(O)	63.17 (15.37)	50.87 (5.77)	78.7 (1.21)
MMDOS(OC)	218.47 (134.34)	36.57 (2.13)	53.57 (0.73)
MMDOS(S)	20 (2.33)	38.83 (2.35)	62.87 (0.86)
MMDOS(SC)	34.13 (6.28)	34 (1.95)	62.3 (2.38)
MMDOS(SO)	**19.43 (1.96)**	38.97 (2.62)	62.57 (0.86)
MMDOS	44.57 (51.5)	33.9 (2.23)	61.5 (2.49)
SCOOT	89.97 (52.73)	38.5 (3.14)	64.57 (3.43)
SCOOT(C)	114.57 (19.11)	53.6 (4.61)	53.93 (0.94)
SCOOT(O)	107.67 (66.5)	51.07 (5.09)	75.87 (1.33)
SCOOT(OC)	159.8 (63.05)	55.23 (5.25)	**53.47 (0.82)**
SCOOT(S)	45.33 (6.57)	**25.13 (1.43)**	59.9 (3.39)
SCOOT(SC)	71.87 (15.93)	41.87 (3.78)	63.9 (2.73)
SCOOT(SO)	41.8 (5.42)	25.37 (1.5)	55.5 (3.13)
SUPRL	46.6 (7.83)	30.13 (1.81)	68.87 (1.55)

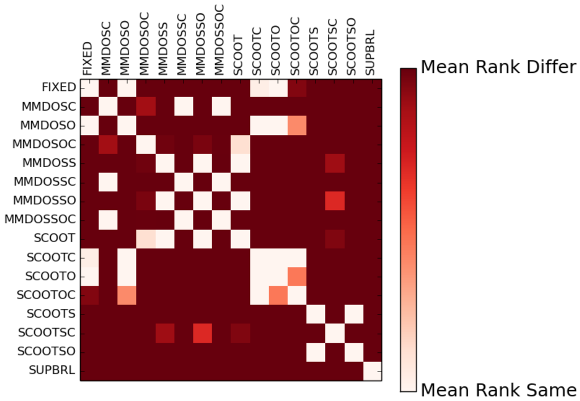
TABLE 7.4: Average *number of stops* (ANS) for each mechanism and traffic scenario.



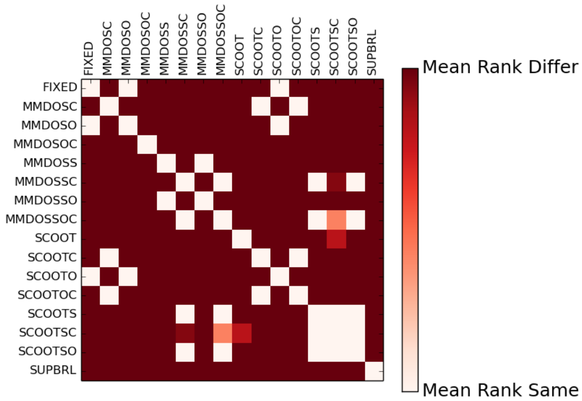
(a) Unstructured.



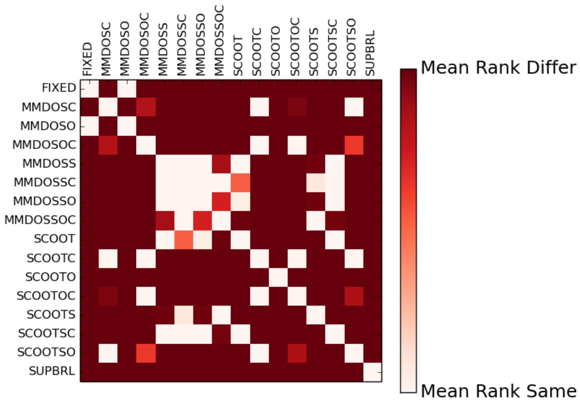
(b) Structured.



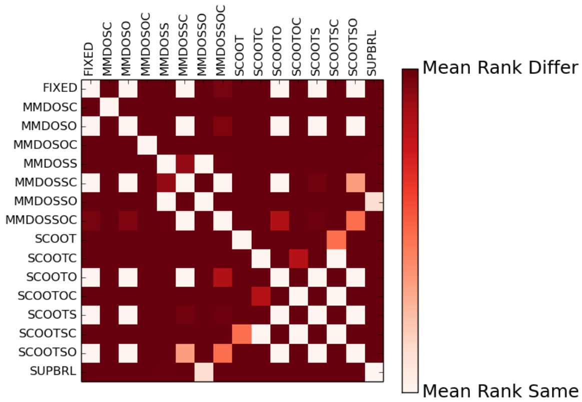
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 7.7: Visual representation of two-sample Mann-Whitney test conducted on ANS (Phoenix map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

7.2.6 Cumulative Average Number of Stops (CANS)

In *unstructured* and *constant* traffic, the CANS results mirror the CATT, that is, the mechanisms with lower CATT also have lower CANS. In *unstructured* traffic, Figure 7.8a, MMDOS(S) and MMDOS(SO) have the lowest CANS throughout the scenario. MMDOS(S) and MMDOS(SO) both have an increase in CANS once the disruption begins, however, both mechanisms maintain lower CANS than the other variants and benchmarks. Additionally, on the Phoenix map, in *unstructured* traffic, MMDOS and MMDOS(SC) have lower CANS than SCOOT, SUPRL and FIXED during and after the disruption. In *directional* traffic, MMDOS(OC) and MMDOS(C) have the lowest CANS, see Figure 7.8e. Also, in *directional* on the Phoenix map, except for MMDOS(O), all the variants have lower CANS than SUPRL. Lastly, in *directional* traffic on the Phoenix map, MMDOS(O) which performs similar to FIXED, has the highest CANS.

In the first disruption of the *football* scenario, all the MMDOS variants have lower CANS than SCOOT, see Figure 7.8a. In the second disruption, MMDOS(O) is the only variant with higher CANS than SCOOT. However, during both disruptions, SUPRL has lower CANS than all of the MMDOS variants. During the football match, SCOOT has the lowest CANS but only MMDOS(O) has higher CANS than SUPRL (during the match MMDOS(O) and FIXED have similar CANS).

In *structured* traffic, MMDOS(OC) consistently has lower CANS than the other MMDOS variants, although, periodically MMDOS(C) has even lower CANS than MMDOS(OC), see Figure 7.8b. Also, in *structured* traffic, MMDOS(O) is the only variant with higher CANS than SUPRL and SCOOT. Although MMDOS and MMDOS(SC) does not have the lowest CATT in *regional* traffic, both mechanisms have the lowest CANS in the *regional* traffic scenario, see Figure 7.8d. Additionally, MMDOS(O) which performs similar to FIXED, has the highest CANS amongst the variants in *regional* traffic. Lastly, in *constant* traffic, MMDOS(OC) and MMDOS(C) have the lowest CANS, see Figure 7.8f. MMDOS(OC) and MMDOS(C) are the only two variants with CANS less than SUPRL, SCOOT and FIXED. All of the other variants have CANS lower than SUPRL but higher than FIXED in *constant* traffic.

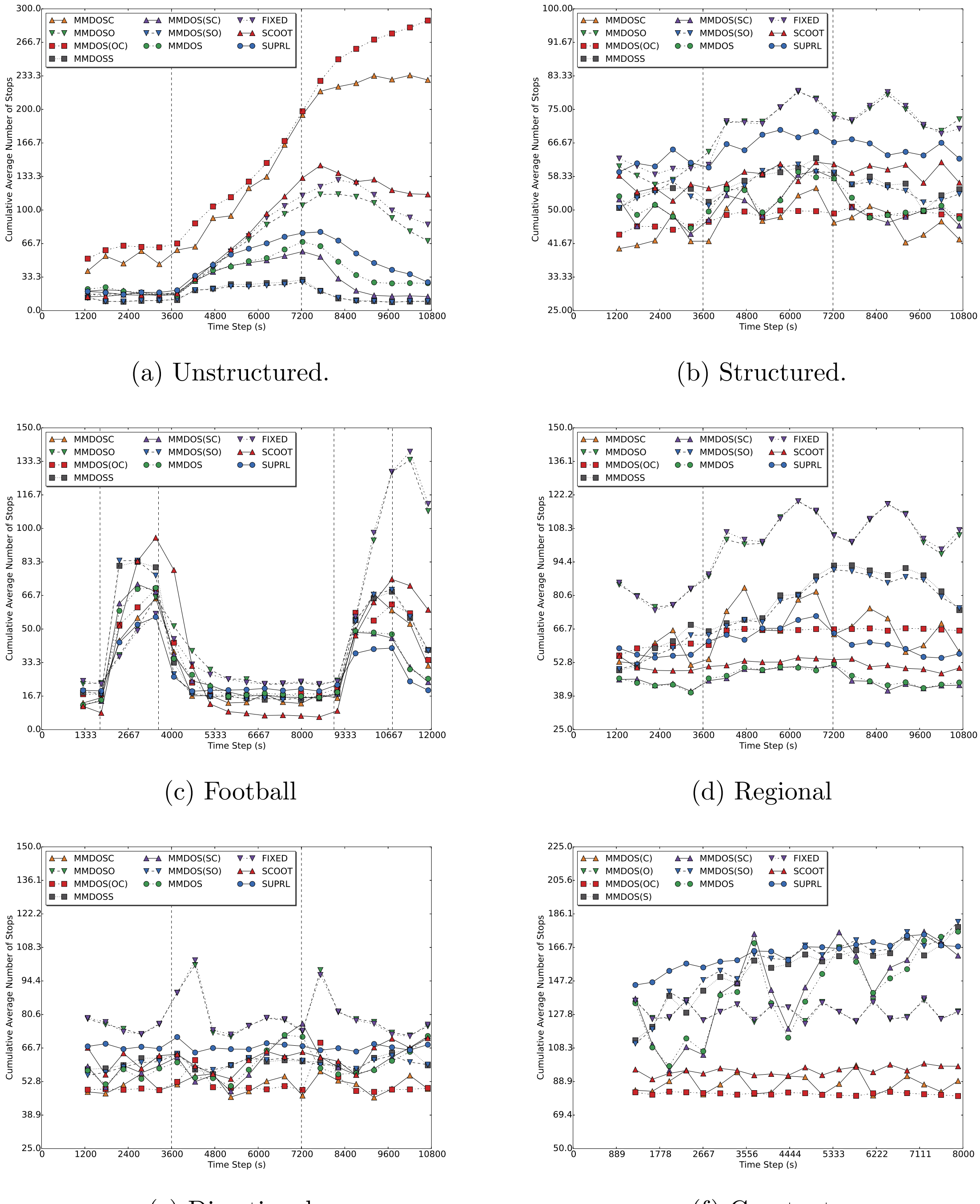
The CANS results for the SCOOT variants in *unstructured* traffic mirror the CATT and CAD results. In *unstructured* traffic, Figure 7.9a, during and after the disruption, SCOOT(SO) has the lowest CANS. In addition, SCOOT(S) has similar CANS to SUPRL in *unstructured*, both mechanisms have lower CANS than SCOOT. SCOOT(SC) is the only other SCOOT variant with lower CANS than SCOOT; SCOOT(O), SCOOT(OC) and SCOOT(C) all have higher CANS than SCOOT, SUPRL and FIXED.

In the *football* scenario, Figure 7.9c, during the first disruption, all the variants except SCOOT(SC) have lower CANS than SCOOT. However, only SCOOT(SO) has lower CANS during the second disruption. During the football match, both SCOOT and SCOOT(SC) have the lowest CANS. Finally, in *directional* traffic, SCOOT(OC) and SCOOT(C) have lower CANS than FIXED, SCOOT and SUPRL, see Figure 7.9e. Also, prior to the *directional* traffic disruption, SCOOT(SO) and SCOOT(S) have similar

CANS to SCOOT and SUPRL, respectively. However, during the disruption the CANS of SCOOT(SO) and SCOOT(S) decreases and remains lower than the CANS of SCOOT and SUPRL for the remainder of the scenario. Lastly, in *directional* traffic, MMDOS(O) is the variant with the highest CANS.

In *structured* traffic, SCOOT(C) and SCOOT(OC) have the lowest CANS, see Figure 7.9b. In addition, just as in *directional* traffic, the CANS of SCOOT(SO) and SCOOT(S) is initially similar to SCOOT, however, after the *structured* traffic disruption the CANS of SCOOT(SO) and SCOOT(S) is lower than SCOOT's CANS. Also, in *structured* traffic, SCOOT(O) has higher CANS than all the other mechanisms during and after the disruption.

In *regional* traffic, SCOOT(SC), SCOOT(SO) and SCOOT(S) have the lowest CANS, however, the CANS of SCOOT is not far behind, see Figure 7.9d. Additionally, the CANS of SCOOT(O) is again similar to FIXED and the highest amongst the variants. On the Phoenix map, SCOOT(OC) and SCOOT(C) have higher CANS than SUPRL and SCOOT. Finally, in *constant* traffic, all the SCOOT variants have lower CANS than SUPRL, see Figure 7.9f. In *constant* traffic, only SCOOT(OC), SCOOT(C) and SCOOT(SC) have CANS lower than FIXED, however, all three perform similar to SCOOT in terms of CANS.



(a) Unstructured.

(b) Structured.

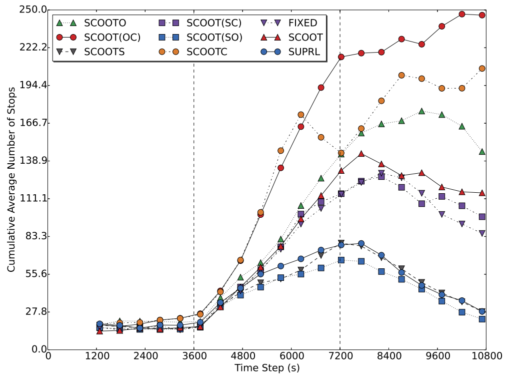
(c) Football

(d) Regional

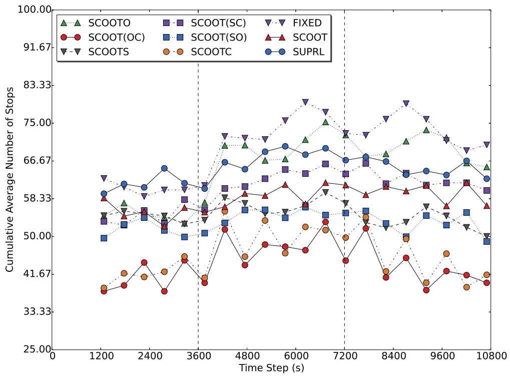
(e) Directional

(f) Constant

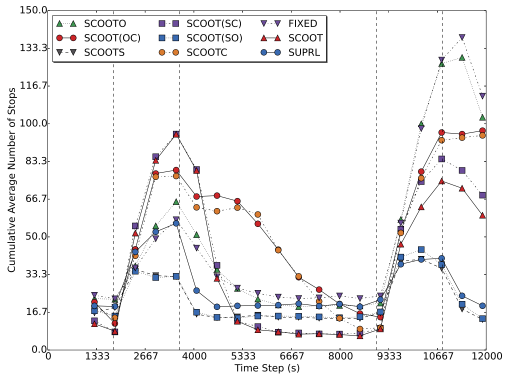
FIGURE 7.8: Cumulative average number of stops (over 30 simulations) on the Phoenix map. Beginning and ending of disruptions are marked by dotted lines.



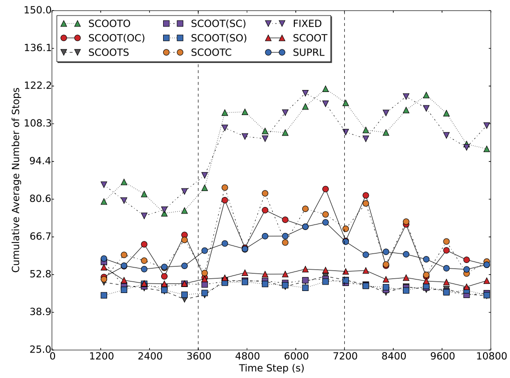
(a) Unstructured.



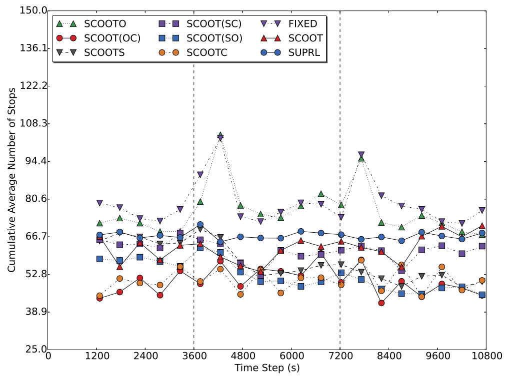
(b) Structured.



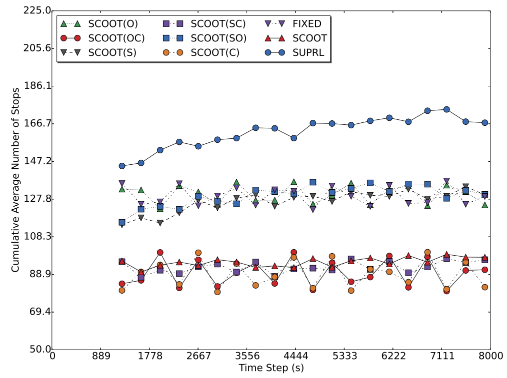
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 7.9: Cumulative average number of stops (over 30 simulations) on the Phoenix map. Beginning and ending of disruptions are marked by dotted lines.

7.3 Results: Portland

This section presents the results of the experiments executed on the Portland map. This section is divided into sub-sections, covering each of the three traffic performance metrics:

average travel time (Section 7.3.1), *traffic density* (Section 7.3.3), and *number of stops* (Section 7.3.5). The traffic control systems are evaluated in three traffic scenarios with *predictable* traffic flow (*structured*, *regional*, and *constant*) and three traffic scenarios with *unpredictable* traffic flow (*unstructured*, *football*, and *directional*). The Mann-Whitney test is used to determine statistical significance between traffic performance results. The threshold value of $p = .05$ was used to determine whether the null hypotheses (the samples were the same) was rejected. The Mann-Whitney test results are presented in a visual manner in lieu of tables to provide the same information but in a more compact manner than a large table(s).

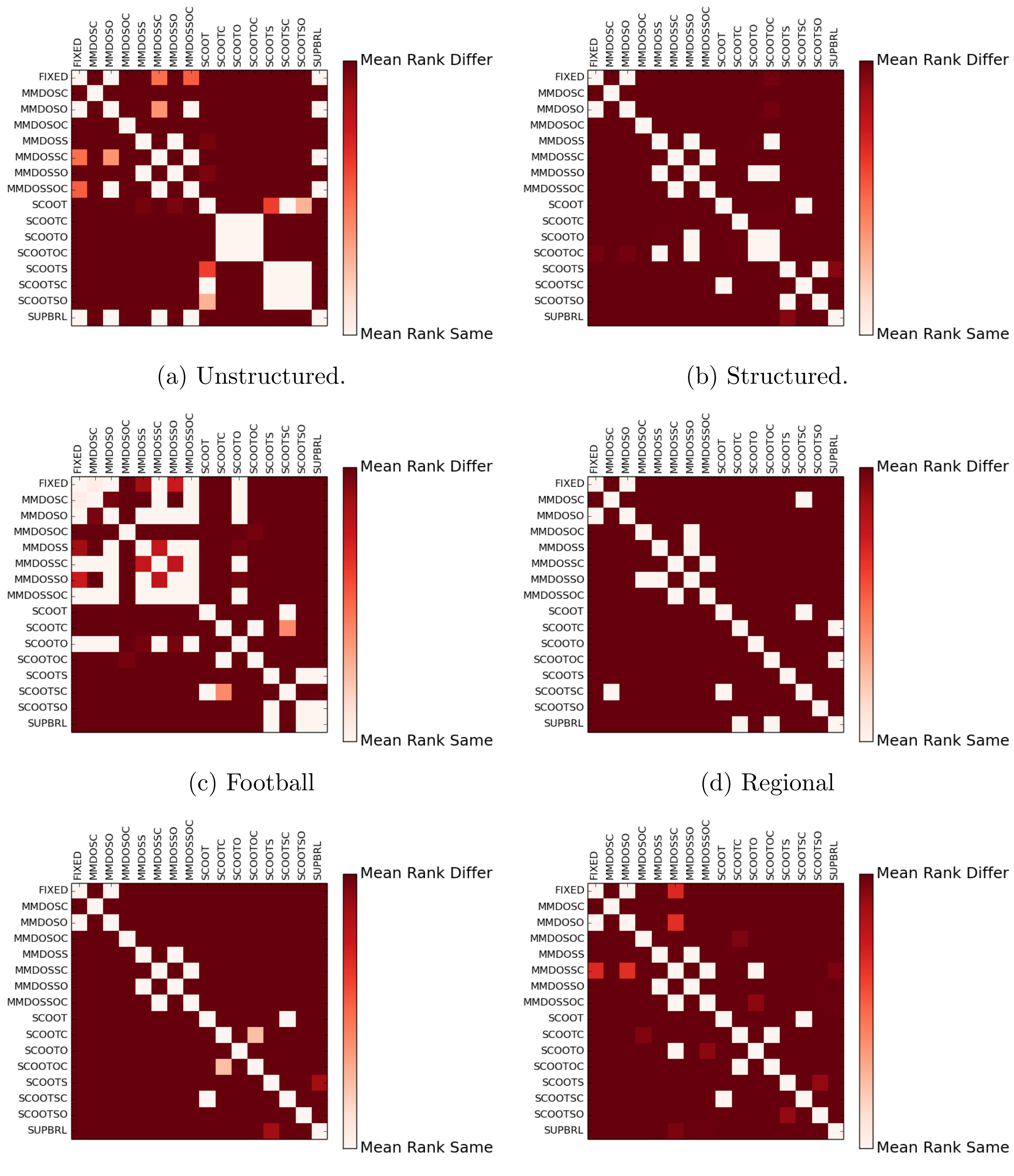
7.3.1 Travel Time (ATT)

In *unstructured* traffic, MMDOS(S) has the overall lowest ATT, Table 7.5. Figure 7.10a shows that MMDOS(S)'s ATT is significantly different from all the other mechanisms except MMDOS(SO). Also, MMDOS(OC) and MMDOS(C) have the top two highest ATT in *unstructured* traffic; this is true in *unstructured* traffic, on the other two maps as well. SCOOT has the lowest ATT amongst the SCOOT variants but the results are not significantly different from SCOOT(S), SCOOT(SC) and SCOOT(SO), see Figure 7.10a. However, in *football* traffic, SUPRL has the lowest ATT which did not occur on the Phoenix map. Additionally, SUPRL performance in the *football* scenario was not significantly different from SCOOT(S) and SCOOT(SO), see Figure 7.10c. MMDOS(SO) and SCOOT(S) have the lowest ATT in the *football* for their respective groups. Although MMDOS(C) performs poorly in *unstructured* traffic, it has the lowest ATT of all the mechanisms in *directional* traffic. Amongst the SCOOT variants, SCOOT(O) has the lowest ATT in *directional* traffic. In comparison to the other mechanisms, the ATT MMDOS(C) and SCOOT(O) in *directional* traffic, is significantly different, see Figure 7.10e.

MMDOS(C) also has the lowest ATT in *structured* and *constant* traffic, Table 7.5. SCOOT(C) and SCOOT(OC) have the lowest ATT amongst the SCOOT variants in *structured* and *constant* traffic. In *regional* traffic, MMDOS(C) has the lowest ATT amongst the MMDOS variants but not overall. SCOOT(S) has the overall lowest ATT in *regional* traffic. MMDOS(C) and SCOOT(S) both have significantly different ATT in *regional* traffic in comparison with SUPRL and FIXED.

	Average Travel Time (ATT) (*std.*)		
		Traffic Pattern	
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	229.38 (0.88)	291.33 (0.66)	265.94 (0.86)
MMDOS(C)	**201.7 (0.58)**	240 (0.42)	**219.75 (0.57)**
MMDOS(O)	229.37 (0.99)	291.29 (0.52)	265.92 (0.81)
MMDOS(OC)	209.79 (0.62)	249.33 (0.46)	223.46 (0.56)
MMDOS(S)	226.91 (1.03)	253.42 (4.5)	240.2 (1.96)
MMDOS(SC)	275.03 (5.1)	273.2 (0.32)	259.95 (9.74)
MMDOS(SO)	226.79 (1.64)	250.89 (4.42)	240.72 (1.37)
MMDOS	274.62 (5.81)	273.12 (0.38)	259.2 (8.16)
SCOOT	360.86 (11)	237.29 (3.97)	286.92 (8.5)
SCOOT(C)	225 (1.55)	245.82 (0.46)	230.55 (13.81)
SCOOT(O)	225.87 (0.79)	290.42 (0.62)	264.62 (0.87)
SCOOT(OC)	227.92 (6.14)	246.73 (1.84)	227.38 (8.52)
SCOOT(S)	241.46 (3.85)	231.51 (2.01)	247.84 (3.77)
SCOOT(SC)	357.45 (12.2)	238.58 (5.97)	288.89 (8.58)
SCOOT(SO)	238.87 (3.76)	**225.66 (2.11)**	244.42 (3.17)
SUPRL	244.13 (1.17)	246.44 (0.94)	253.12 (1.47)
	Unstructured	*Football*	*Directional*
FIXED	569.87 (18.05)	299.6 (5.09)	242.97 (0.88)
MMDOS(C)	855.39 (79.42)	304.58 (8.7)	**207.71 (0.56)**
MMDOS(O)	572.44 (19.66)	297.32 (5.23)	242.82 (0.78)
MMDOS(OC)	1462.05 (217.26)	342.44 (6.64)	216.22 (0.52)
MMDOS(S)	**497.39 (58.79)**	294.07 (9.34)	221.55 (0.84)
MMDOS(SC)	586.13 (21.14)	304.26 (13.91)	270.55 (11.76)
MMDOS(SO)	500.57 (59.16)	293.73 (8.62)	221.67 (0.95)
MMDOS	584.14 (14.98)	299.94 (19.28)	270.31 (9.08)
SCOOT	510.6 (26.52)	392.73 (23.55)	369.13 (4.96)
SCOOT(C)	624.69 (27.6)	361.28 (13.22)	296.5 (7.84)
SCOOT(O)	607.65 (15.45)	301.39 (6.01)	241.32 (0.79)
SCOOT(OC)	625.94 (31.68)	351.99 (12.35)	294.07 (4.32)
SCOOT(S)	526.14 (17.64)	277.92 (16.21)	250.63 (2.51)
SCOOT(SC)	517.92 (34.22)	379.53 (27.41)	362.92 (13.21)
SCOOT(SO)	523.12 (19.49)	278.9 (15.68)	246.82 (2.66)
SUPRL	577.7 (10)	**271.56 (5.62)**	249.16 (1.28)

TABLE 7.5: Average travel times (ATT) for each mechanism and traffic scenario.



(a) Unstructured. (b) Structured.

(c) Football (d) Regional

(e) Directional (f) Constant

Figure 7.10: Visual representation of two-sample Mann-Whitney test conducted on ATT (Portland map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

7.3.2 Cumulative Average Travel Time (CATT)

In the *unstructured* traffic scenario, during the disruption MMDOS(S) and MMDOS(SO) have the lowest CATT, see Figure 7.11a. In *unstructured* traffic, MMDOS(S) and MMDOS(SO) performs similar to SCOOT in terms of CATT. Although prior to the disruption, MMDOS and MMDOS(SC) have higher CATT than the benchmarks, during and after the disruption both mechanisms perform in a similar fashion to SUPRL and FIXED. In addition, the CATT of MMDOS(O) is nearly identical to FIXED. On the Portland map, the *unstructured* traffic disruption does not cause quite an increase in the CATT MMDOS, MMDOS(SC) and MMDOS(O) compared with the *unstructured* traffic disruption on the Phoenix map. However, on both maps, MMDOS(OC) and MMDOS(C) have the highest and second highest CATT, respectively, in *unstructured* traffic.

In the *football* traffic scenario, during the first disruption, all of the MMDOS variants behave in a similar manner, see Figure 7.11c. However, during the match MMDOS(O) which performs similar to FIXED, has lower CATT than the other MMDOS variants. MMDOS(O) also has the lower CATT than the other MMDOS variants during and after the second disruption. The other MMDOS mechanisms perform as well as or better than SCOOT in the *football* scenario. Additionally, MMDOS(S) and MMDOS(SO) perform in a similar manner to SUPRL in the *football* scenario. Lastly, in the *football* scenario, Figure 7.11c shows that MMDOS(OC) peak CATT is greatest during the football match and during that same period, exceeded the CATT of SCOOT. On the Phoenix map, during the same period in the *football* scenario, MMDOS(OC) has lower CATT than FIXED and SCOOT.

However, in *directional* traffic, MMDOS(OC) and MMDOS(C) have the lowest CATT, see Figure 7.11e. MMDOS(OC) and MMDOS(C) also have the lowest CATT in *directional* traffic on the Phoenix map. In *directional* traffic, MMDOS(S) and MMDOS(SO), have lower CATT than FIXED, SCOOT and SUPRL. Also, in *directional* traffic, the CATT of MMDOS(SC) is nearly identical to MMDOS. Lastly, in *directional* traffic, MMDOS(O) performs similar to FIXED (both methods have lower CATT) than SCOOT and SUPRL. On the Phoenix map, MMDOS(O) also performs similar to FIXED but the two methods have the highest CATT in *directional* traffic.

In *structured* and *constant* traffic MMDOS(OC) and MMDOS(C) have the lowest CATT, Figure 7.11b and 7.11f. This is also true in *structured* and *constant* traffic on the Phoenix map. Also in *structured* and *constant* traffic, MMDOS(S) and MMDOS(SO) have lower CATT than SUPRL and SCOOT. However, in *structured* traffic, MMDOS(O) which performs similar to FIXED, has lower CATT than SUPRL and SCOOT but in *constant* traffic, MMDOS(O) has CATT than SUPRL and SCOOT (during the second half of the scenario).

Prior to the *regional* traffic disruption, Figure 7.11d, SCOOT has the lowest CATT, however, during the disruption MMDOS(C) and SCOOT have similar CATT. After the disruption has ended, SCOOT and MMDOS(OC) have similar CATT and MMDOS(C) has the lowest CATT. Although prior to the disruption, MMDOS(S) and MMDOS(SO) have lower CATT than SUPRL, during and after the disruption, MMDOS(S) and MMDOS(SO) behaves similar to SUPRL. Lastly, in *regional* traffic, all the MMDOS(O) has the highest CATT amongst the variants and behaves similar to FIXED.

In *unstructured* traffic, SCOOT(SC) and SCOOT have the lowest CATT prior to and during the *unstructured* traffic scenario, see Figure 7.12a. However, once the disruption has ended SCOOT(S), SCOOT(SO), SCOOT(SC) and SCOOT behave in a similar manner. In addition, in *unstructured* traffic, prior to the disruption, SCOOT(O), SCOOT(C) and SCOOT(OC) have similar CATT to FIXED and SUPRL. Lastly, SCOOT(O), SCOOT(C) and SCOOT(OC) have higher CATT than FIXED, SUPRL and SCOOT during and after the disruption. SCOOT(C) and SCOOT(OC) also have higher CATT than the other SCOOT variants in *unstructured* traffic on the Phoenix map.

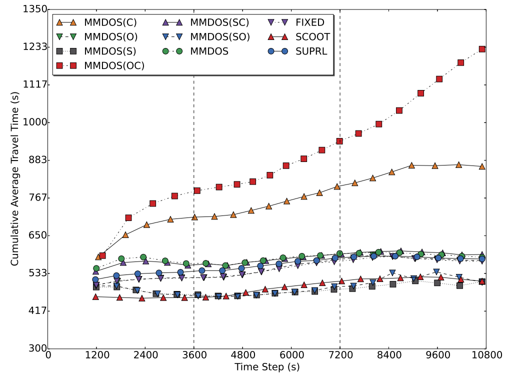
In the *football* traffic scenario, excluding SCOOT(C) and SCOOT(OC), all the SCOOT variants and benchmarks have a similar increase in CATT, see Figure 7.12c. However, during the football match, SCOOT(O), SCOOT(SO), SCOOT(S) have similar CATT to FIXED but have lower CATT than SCOOT and SUPRL. SCOOT(O), SCOOT(SO), SCOOT(S) continue to have lower CATT than SUPRL and SCOOT during the second *football* traffic disruption. SCOOT(C) and SCOOT(OC) have higher CATT than the other SCOOT variants during the first disruption and the football match. During the second disruption, SCOOT(C), SCOOT(OC), SCOOT(SC) and SCOOT have higher CATT than SUPRL and FIXED.

In *directional* traffic, SCOOT(O) performs similar to FIXED, however, the two methods have the lowest CATT, see Figure 7.12e. Also, in *directional* traffic, SCOOT(S) and SCOOT(SO) have similar CATT to SUPRL. Figure 7.12e, shows that the CATT of SCOOT(S), SCOOT(SO), and SCOOT(O) changes little during the *directional* traffic disruption. In contrast, the CATT of SCOOT(C), SCOOT(OC), and SCOOT(SC) increases during the *directional* traffic disruption. Also, during the *directional* scenario on the Phoenix map, SCOOT(C) and SCOOT(OC) have the lowest CATT.

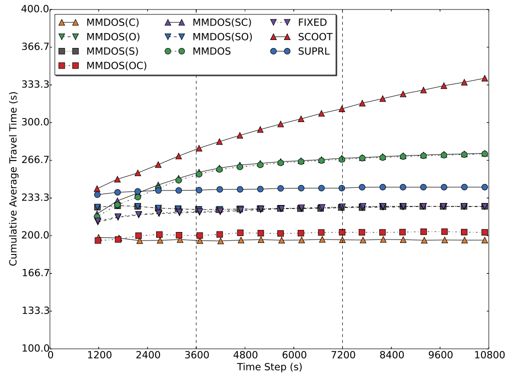
Although initially, SCOOT(C) and SCOOT(OC) have higher CATT than FIXED and SCOOT(O) in *structured* traffic, during and after the *structured* traffic disruption, all four mechanisms have similar CATT which is lower than SUPRL and SCOOT, see Figure 7.12b. In *structured* traffic, SCOOT(S) and SCOOT(SO) performs similar to SUPRL. Additionally, only SCOOT(SC) and SCOOT have an increase in CATT during the *structured* traffic disruption.

In *regional* traffic, SCOOT(SO) has the lowest CATT beginning mid-disruption and after the disruption ended, see Figure 7.12d. Prior to the disruption, SCOOT(SO) has higher CATT than SCOOT. Figure 7.12d shows that in *regional* traffic the SCOOT variants show little change in CATT during the disruption and that only SCOOT(O) has higher CATT than SCOOT and SUPRL.

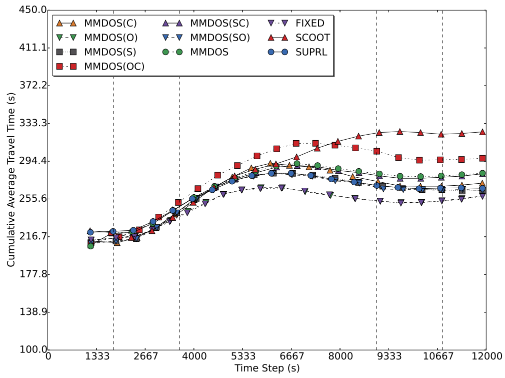
Lastly, in *constant* traffic, SCOOT(C) and SCOOT(OC) have the lowest CATT, see Figure 7.12f. Also, in *constant* traffic, SCOOT(S) and SCOOT(SO) have lower CATT than SUPRL and outperforms SCOOT as the scenario progresses. Only SCOOT(SC) (and SCOOT) have an increase in CATT as the simulation runs. Finally, Figure 7.12f, also shows that SCOOT(O) and FIXED have similar CATT in *constant* traffic.



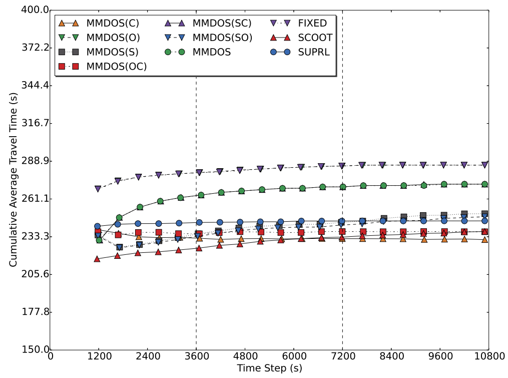
(a) Unstructured.



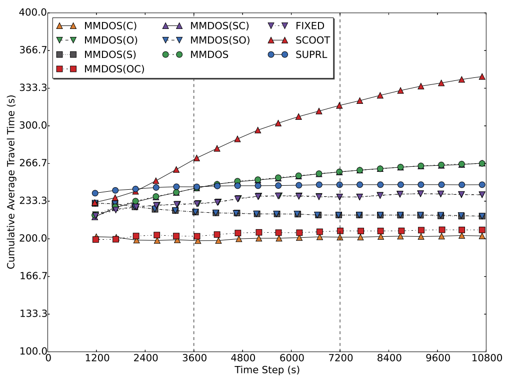
(b) Structured.



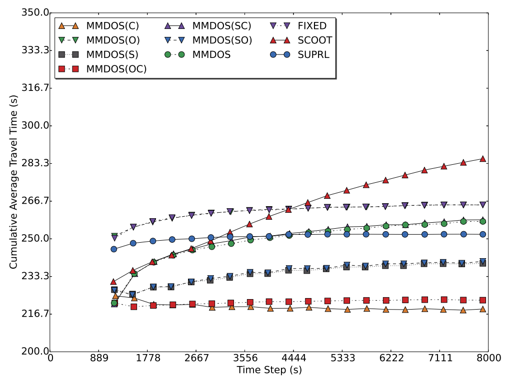
(c) Football



(d) Regional

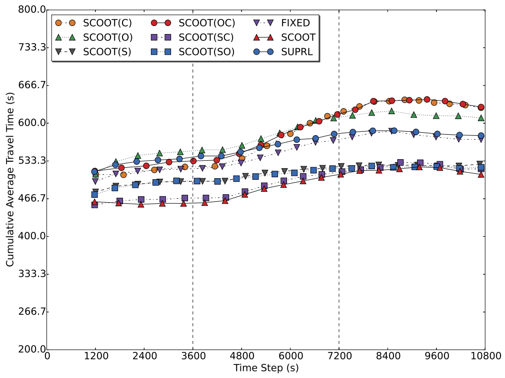


(e) Directional

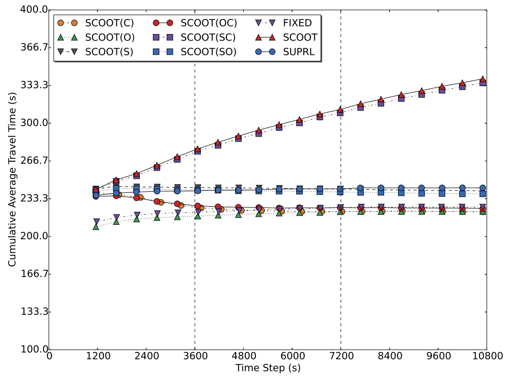


(f) Constant

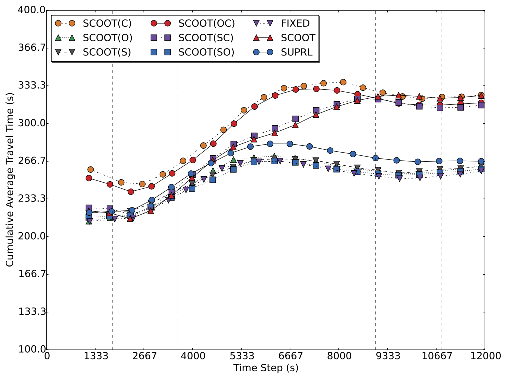
Figure 7.11: Cumulative average travel times (over 30 simulations) on the Portland map. Beginning and ending of disruptions are marked by dotted lines.



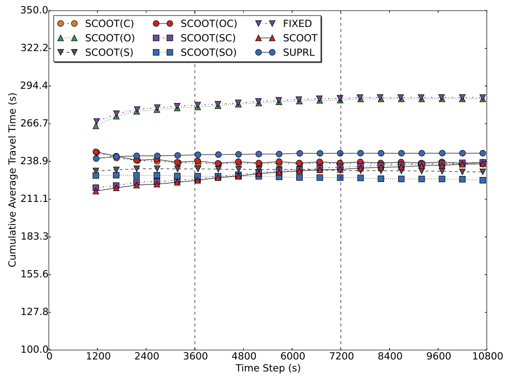
(a) Unstructured.



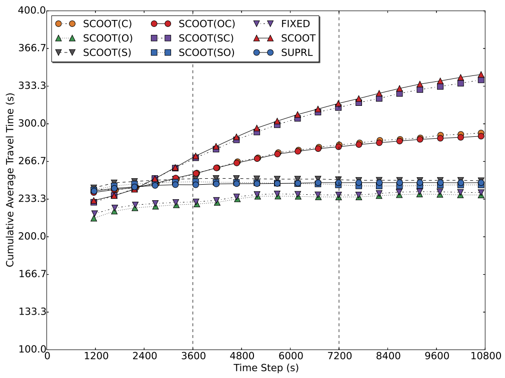
(b) Structured.



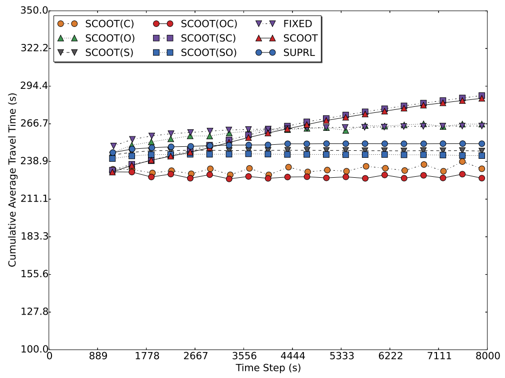
(c) Football



(d) Regional



(e) Directional



(f) Constant

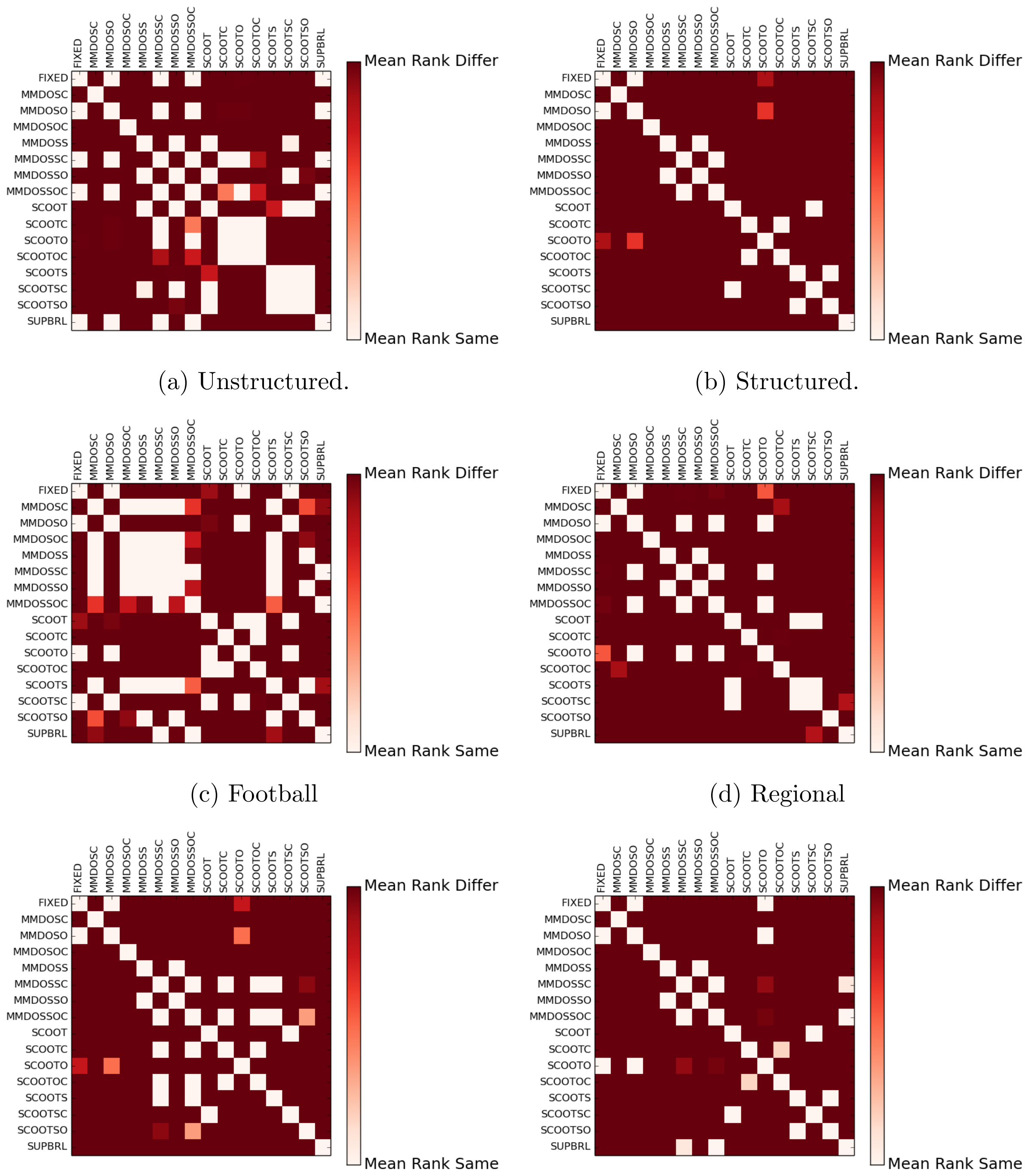
FIGURE 7.12: Cumulative average travel times (over 30 simulations) on the Portland map. Beginning and ending of disruptions are marked by dotted lines.

7.3.3 Density (ATD)

Table 7.6 shows that MMDOS(S) has the lowest ATD in *unstructured* traffic but it is not significantly different from MMDOS(SO), SCOOT and SCOOT(SC). MMDOS(S) also has the lowest ATD in *football* traffic but only amongst the MMDOS variants. SCOOT(S) has the over all lowest ATD in *football* traffic but, amongst the MMDOS variants, it is only significantly different from MMDOS(O), see Figure 7.13c. Figure 7.13c also shows that in *football* traffic, only MMDOS(O) performs significantly different from the other MMDOS variants. In *directional*, MMDOS(OC) has the lowest overall ATD and the results are significantly different from the other MMDOS variants, see Figure 7.13e. MMDOS(OC) also has the overall lowest ATD in *structured*, *regional* and *constant* traffic; in all three scenarios the results are significantly different from the other mechanisms. SCOOT(C) has the lowest ATD amongst the SCOOT variants in *structured* traffic, however, it is not significantly different from SCOOT(OC). In *constant* and *regional* traffic, SCOOT(OC) has the lowest ATD in the SCOOT group.

Average Traffic Density (ATD) (*std.*)			
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	12.42 (0.21)	20.46 (0.2)	20.69 (0.16)
MMDOS(C)	9.11 (0.12)	13.48 (0.13)	15.81 (0.06)
MMDOS(O)	12.39 (0.16)	20.35 (0.27)	20.65 (0.17)
MMDOS(OC)	**8.76 (0.13)**	**12.18 (0.12)**	**15.32 (0.06)**
MMDOS(S)	13.45 (0.14)	18.62 (0.48)	18.52 (0.18)
MMDOS(SC)	15.6 (0.37)	20.15 (0.33)	19.88 (0.71)
MMDOS(SO)	13.42 (0.2)	18.33 (0.42)	18.54 (0.19)
MMDOS	15.39 (0.57)	20.25 (0.2)	19.8 (0.63)
SCOOT	18.24 (0.58)	16.98 (0.39)	22.66 (0.75)
SCOOT(C)	10.19 (0.19)	13.8 (0.18)	16.67 (1.01)
SCOOT(O)	12.25 (0.16)	20.25 (0.28)	19.25 (3.97)
SCOOT(OC)	10.23 (0.26)	13.62 (0.18)	16.36 (0.63)
SCOOT(S)	14.44 (0.26)	16.86 (0.25)	19.1 (0.36)
SCOOT(SC)	18.01 (0.65)	17.02 (0.52)	22.79 (0.79)
SCOOT(SO)	14.31 (0.24)	16.45 (0.23)	18.86 (0.3)
SUPRL	13.94 (0.23)	17.39 (0.2)	19.52 (0.19)
	Unstructured	*Football*	*Directional*
FIXED	2.76 (0.14)	9.54 (0.15)	14.03 (0.18)
MMDOS(C)	4.15 (0.38)	8.7 (0.28)	10.93 (0.14)
MMDOS(O)	2.77 (0.16)	9.5 (0.15)	14.01 (0.14)
MMDOS(OC)	6.09 (1.06)	8.71 (0.17)	**10.15 (0.15)**
MMDOS(S)	**2.41 (0.31)**	8.66 (0.25)	13.32 (0.16)
MMDOS(SC)	2.84 (0.14)	8.9 (0.34)	15.55 (0.57)
MMDOS(SO)	2.44 (0.35)	8.68 (0.27)	13.37 (0.15)
MMDOS	2.84 (0.12)	8.92 (0.29)	15.43 (0.71)
SCOOT	2.43 (0.16)	9.94 (0.52)	19.24 (0.38)
SCOOT(C)	2.97 (0.19)	10.51 (0.47)	15.72 (0.5)
SCOOT(O)	2.93 (0.15)	9.59 (0.18)	13.87 (0.19)
SCOOT(OC)	3 (0.2)	10.19 (0.51)	15.7 (0.29)
SCOOT(S)	2.53 (0.11)	8.6 (0.52)	15.27 (0.23)
SCOOT(SC)	2.48 (0.21)	9.7 (0.57)	18.77 (0.85)
SCOOT(SO)	2.5 (0.12)	**8.46 (0.44)**	15.05 (0.16)
SUPRL	2.78 (0.11)	8.96 (0.26)	14.76 (0.13)

Table 7.6: Average traffic density (ATD) for each mechanism and traffic scenario.



(a) Unstructured. (b) Structured.

(c) Football (d) Regional

(e) Directional (f) Constant

Figure 7.13: Visual representation of two-sample Mann-Whitney test conducted on ATD (Portland map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

7.3.4 Cumulative Average Density (CAD)

In *unstructured* traffic, the majority of the MMDOS variants behave in a similar manner in terms of CAD, see Figure 7.14a. Prior to and during the *unstructured* traffic disruption, SCOOT, MMDOS(S) and MMDOS(SO) have the lowest CAD. Once the disruption ended, all the MMDOS variants except MMDOS(OC) and MMDOS(C) have similar CAD. The CAD of MMDOS and MMDOS(SC) is nearly identical to the CAD of SUPRL and FIXED. MMDOS(OC) and MMDOS(C) have higher CAD than the other MMDOS variants and are the only variants that have an increase in CAD during the entire disruption period.

In the first disruption in the *football* scenario, all the MMDOS variants have a similar increase in CAD, although, MMDOS(O) (and FIXED) achieve the lowest CAD during that period, see Figure 7.14c. During the football match, MMDOS(O) and FIXED has the lowest CAD. Additionally, during the football match, MMDOS(S) and MMDOS(SO) have similar CAD to SUPRL, both variants have lower CAD than SCOOT. During the second disruption, all of the MMDOS variants have lower CAD than SCOOT, however, SUPRL have the lowest CAD in comparison to the MMDOS variants.

In *directional* traffic on the Portland map, all the MMDOS variants have lower CAD than SCOOT, see Figure 7.14e. MMDOS(OC) and MMDOS(C) have the lowest CAD in the *directional* traffic scenario and display little change during the disruption. Additionally, in *directional* traffic, MMDOS(S) and MMDOS(SO) have lower CAD than all three benchmarks. Although MMDOS and MMDOS(SC) have lower CAD than SCOOT, both mechanisms have higher CAD than SUPRL. Lastly, in *directional* traffic, the CAD of MMDOS(O) is nearly identical to the CAD of FIXED.

Figure 7.14 shows that in all three scenarios with *predictable* traffic, MMDOS(OC) and MMDOS(C) have the lowest CAD. On the Phoenix map, MMDOS(OC) and MMDOS(C) have the lowest CAD in *structured* and *constant* traffic. In *structured* traffic, MMDOS(S), MMDOS(SO) and MMDOS(O) have similar CAD to FIXED, see Figure 7.14b. In addition, MMDOS and MMDOS(SC) have lower CAD than SCOOT but greater than FIXED and SUPRL.

Figure 7.14d shows that in *regional* traffic, only MMDOS(OC) and MMDOS(C) have lower CAD than all three benchmarks. In *regional* traffic, MMDOS, MMDOS(SC), MMDOS(S) and MMDOS(SO) have lower CAD than FIXED but higher than SUPRL and SCOOT.

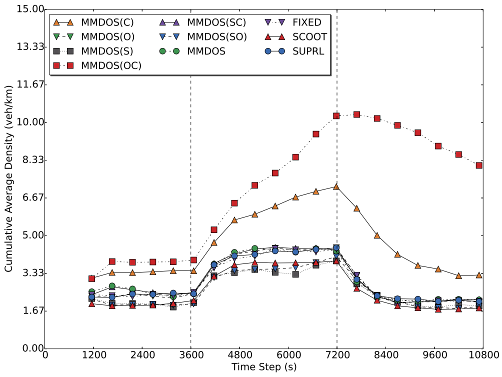
In *constant* traffic, soon after the scenario begins SCOOT has higher CAD than all of the MMDOS variants. Additionally, all of the MMDOS variants, except MMDOS(O) have lower CAD than FIXED. Lastly, in all scenarios with *predictable* traffic, MMDOS(O) and FIXED have nearly identical CAD.

In *unstructured* traffic, the SCOOT variants display little difference in terms of CAD, see Figure 7.15a. The greatest difference amongst the SCOOT variants occurs during the disruption, where SCOOT(S), SCOOT(SO), and SCOOT(SC) which perform in a similar fashion to SCOOT have the lowest CAD amongst the SCOOT variants. During and after the *unstructured* traffic disruption, SCOOT(C) and SCOOT(OC) have the highest CAD.

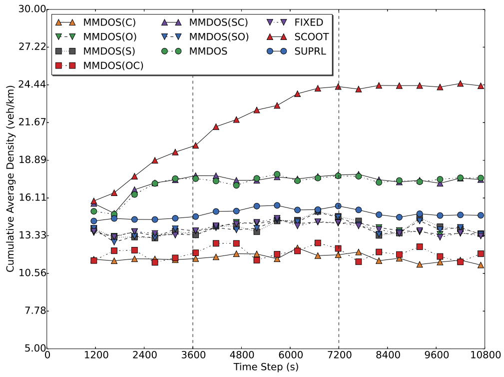
In the *football* scenario, during the first disruption all of the mechanisms have a similar increase in CAD, see Figure 7.15c. Towards the end of the first disruption, SCOOT(SO), SCOOT(S), SCOOT(O) perform as well as FIXED which has the lowest CAD. During the football match, SCOOT(SO), SCOOT(S), SCOOT(O) have the lowest CAD amongst the SCOOT variants, however, their performance is similar to FIXED. Additionally, during and after the football match, all the SCOOT variants perform as well as or better than SCOOT. During the second *football* disruption, SCOOT(SO) and SCOOT(S) have lower CAD than FIXED and SCOOT but not SUPRL.

In the *directional* traffic scenario, all of the SCOOT variants, excluding SCOOT(SC), have lower CAD than SCOOT, see Figure 7.15e. FIXED and SCOOT(O) have the lowest CAD during certain portions of the *directional* traffic scenario, e.g., prior to the disruption and during the end of the disruption, see Figure 7.15e. Also in *directional* traffic, the CAD of SCOOT(S) and SCOOT(SO) is similar to the CAD of SUPRL. On the Phoenix map, SCOOT(OC) and SCOOT(C) have the lowest CAD amongst the SCOOT variants in *directional* traffic, however, on the Portland map, SCOOT(OC) and SCOOT(C) have lower CAD than SCOOT but not FIXED and SUPRL.

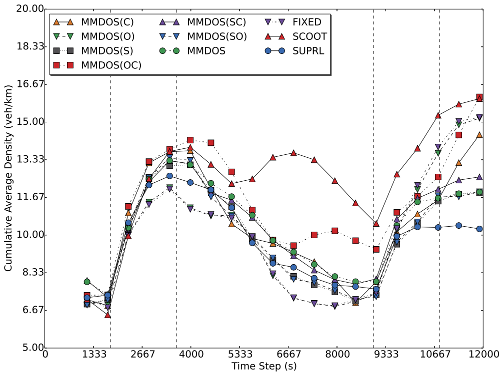
SCOOT(OC) and SCOOT(C) have the lowest CAD in all three scenarios with *predictable* traffic. In *structured* traffic, SCOOT(SC) and SCOOT which have nearly identical CAD, have the highest CAD, see Figure 7.15b. Additionally, other than SCOOT(OC) and SCOOT(C), in *structured* traffic, SCOOT(O) which performs similar to FIXED is the only other SCOOT variant with lower CAD than SCOOT and FIXED. In *regional* traffic, excluding SCOOT(O), all of the SCOOT variants perform as well as or better than the benchmarks, see Figure 7.15d. SCOOT(O) (and FIXED) have the highest CAD in *regional* traffic. Figure 7.15f, shows that SCOOT(S), SCOOT(SO) and SCOOT(O) have similar CAD to SUPRL in *constant* traffic. In addition, SCOOT(SC) is the only SCOOT variant with greater CAD than FIXED and SUPRL. SCOOT(SC) and SCOOT have the highest CAD in *structured* and *constant* traffic, in comparison to the other SCOOT variants.



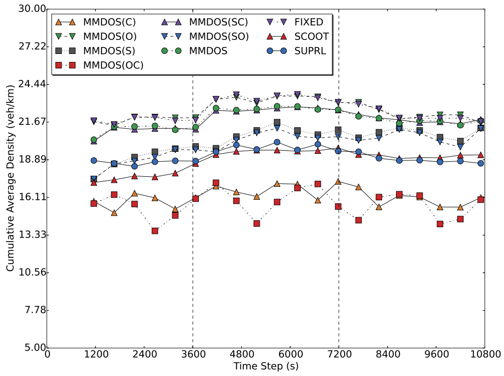
(a) Unstructured.



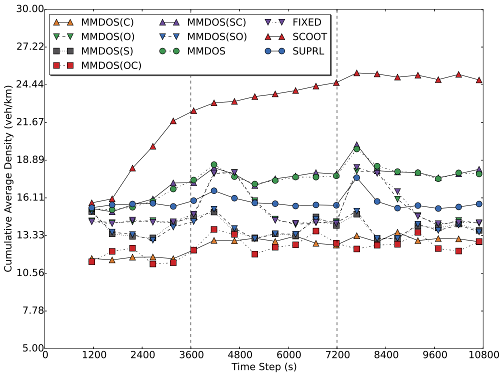
(b) Structured.



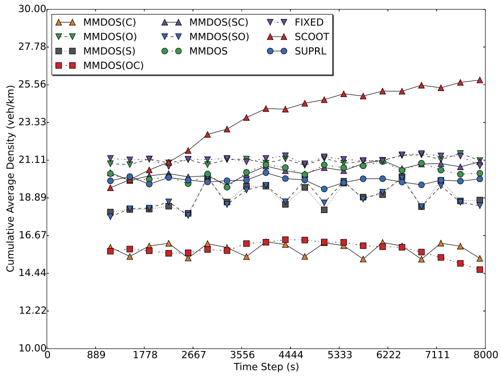
(c) Football



(d) Regional

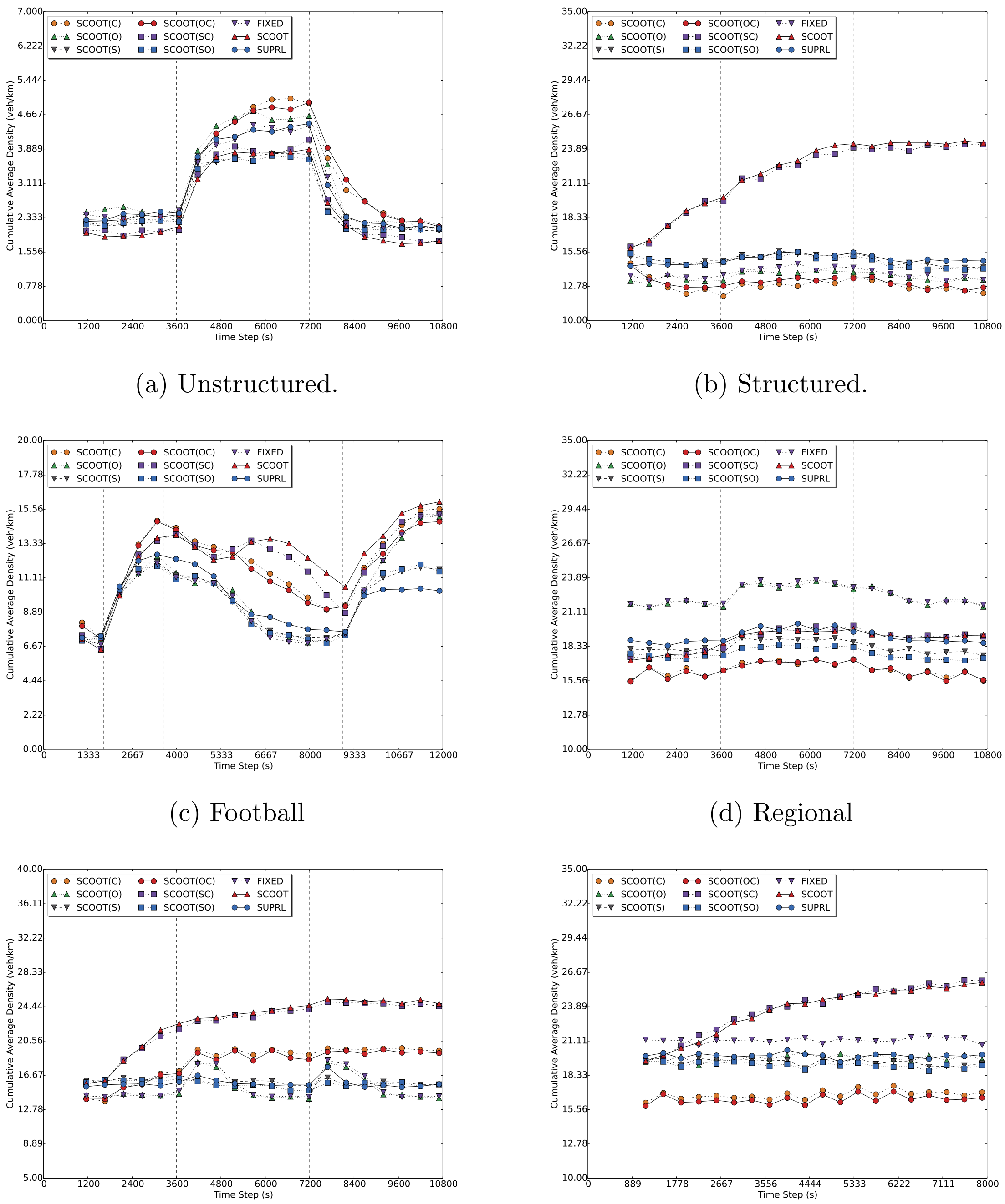


(e) Directional



(f) Constant

FIGURE 7.14: Cumulative average density (over 30 simulations) on the Portland map. Beginning and ending of disruptions are marked by dotted lines.



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

(e) Directional

(f) Constant

FIGURE 7.15: Cumulative average density (over 30 simulations) on the Portland map. Beginning and ending of disruptions are marked by dotted lines.

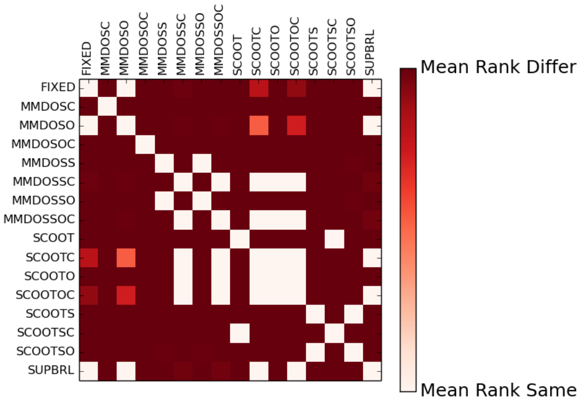
7.3.5 Vehicle Stops (ANS)

Amongst the MMDOS variants, MMDOS(S) has the lowest ANS in *unstructured* traffic. However, SCOOT has the overall lowest ANS in *unstructured* traffic; Figure 7.16a shows that it is not significantly different from SCOOT(SC) in *unstructured* traffic. The Portland map is the only map where SCOOT outperforms all the other mechanisms in ANS with *unstructured* traffic. In *football* traffic, SCOOT(SO) has the overall lowest ANS, however, this is not significantly different from SCOOTS, SCOOTSO and SUPRL, see Figure 7.16c. Amongst the MMDOS variants, MMDOS(SC) has the lowest ANS, however, its performance is not significantly different from MMDOS(S), MMDOS(SO), and MMDOS(SOC), see Figure 7.16c. In *directional* traffic, MMDOS(OC) has the overall lowest ANS. SCOOT(SO) has the lowest ANS in *directional* within the SCOOT group.

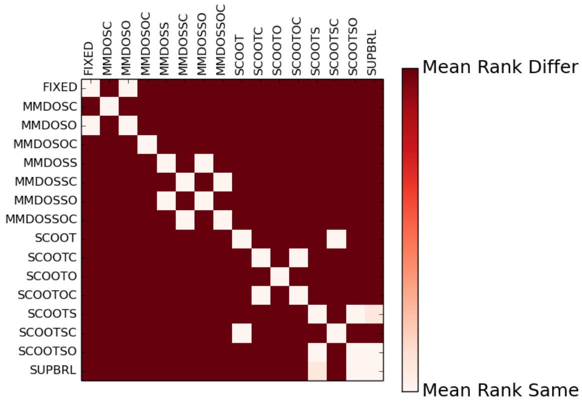
In the *structured* traffic scenario, the *cycle* only mechanisms have the lowest ANS. MMDOS(C) and SCOOT(C) have the lowest ANS in their respective groups in *structured* traffic. However, SCOOT(C) has the overall lowest ANS which is significantly different from all the other mechanisms except SCOOT(OC), see Figure 7.16b. In *regional* traffic, MMDOS(OC) and SCOOT(SO) have the lowest ANS in their respective groups; SCOOT(SO) has the overall lowest ANS. Figure 7.16d shows that both mechanisms have significantly different ANS when in comparison to all the other mechanisms in *regional* traffic. In *constant* traffic, MMDOS(SC) has the overall lowest ANS and SCOOT(OC) has the lowest ANS in the SCOOT group. In all three scenarios with predictable traffic, *structured*, *regional* and *constant*, the variants outperform the benchmarks in preventing stops.

Average Number of Stops (ANS) (*std.*)			
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	161.13 (2.71)	280.5 (2.71)	279.97 (2.54)
MMDOS(C)	122.8 (1.61)	187.27 (1.95)	**217.2 (0.89)**
MMDOS(O)	160.67 (2.15)	279 (3.72)	279.3 (2.69)
MMDOS(OC)	125.47 (1.89)	179.73 (1.8)	222.2 (0.96)
MMDOS(S)	170.13 (2.19)	239.2 (10.22)	231.83 (4.13)
MMDOS(SC)	226.2 (6.89)	203.07 (3.63)	226.4 (9.4)
MMDOS(SO)	169.87 (3.13)	233 (8.79)	232.53 (3.46)
MMDOS	222.6 (10.41)	203.8 (2.41)	226.03 (8.29)
SCOOT	319.8 (13.78)	203.77 (7.74)	347.6 (19.31)
SCOOT(C)	**118.8 (2.68)**	193.83 (2.55)	236.33 (24.1)
SCOOT(O)	156.67 (2.04)	276.17 (3.92)	258.3 (54.71)
SCOOT(OC)	120.47 (6.56)	191.83 (2.87)	230.07 (14.9)
SCOOT(S)	189.83 (6.92)	186.43 (5.07)	240.43 (8.17)
SCOOT(SC)	314 (15.13)	204.57 (11.87)	350.7 (20.55)
SCOOT(SO)	185.73 (5.87)	**175.1 (4.17)**	233.73 (6.71)
SUPRL	185.33 (3.18)	201.1 (2.68)	246 (3.45)
	Unstructured	*Football*	*Directional*
FIXED	28.83 (2.45)	143.4 (3.61)	187.1 (2.5)
MMDOS(C)	69.63 (9.82)	147.97 (6.67)	149.6 (1.92)
MMDOS(O)	29.17 (2.76)	141.87 (3.5)	186.83 (1.91)
MMDOS(OC)	133.67 (37.45)	161 (4.09)	**147.67 (2.31)**
MMDOS(S)	23.73 (7.39)	129.73 (6.12)	162.93 (2.41)
MMDOS(SC)	31.93 (2.73)	128.8 (8.84)	211.83 (10.91)
MMDOS(SO)	24.23 (7.83)	129.8 (5.91)	163.63 (2.43)
MMDOS	31.8 (2.01)	127.57 (9.9)	209.53 (12.27)
SCOOT	**19.27 (2.27)**	153.77 (13.37)	344.93 (7.87)
SCOOT(C)	31.63 (3.47)	145.7 (9.21)	241.37 (11.81)
SCOOT(O)	32.97 (2.39)	144.53 (4.35)	183.37 (2.63)
SCOOT(OC)	31.77 (3.78)	138.8 (9.85)	239.23 (6.87)
SCOOT(S)	25.5 (2.06)	**120.9 (13.05)**	203.6 (4.74)
SCOOT(SC)	20.1 (3.35)	145.37 (15.16)	332.7 (20.06)
SCOOT(SO)	25.2 (2.23)	119.07 (11.77)	196.73 (4.01)
SUPRL	29.63 (1.52)	122.73 (4.89)	194.37 (2.34)

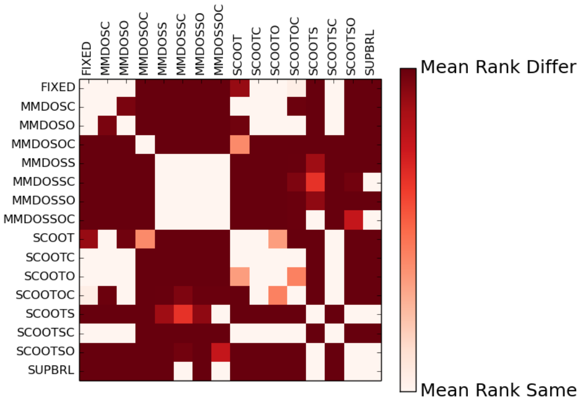
TABLE 7.7: Average *number of stops* (ANS) for each mechanism and traffic scenario.



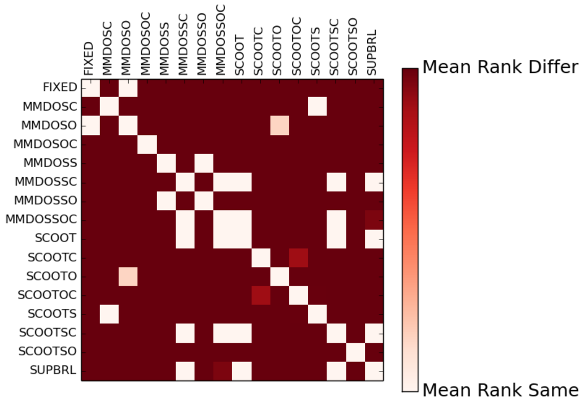
(a) Unstructured.



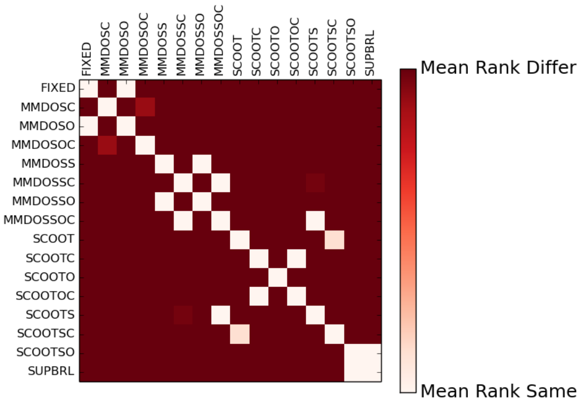
(b) Structured.



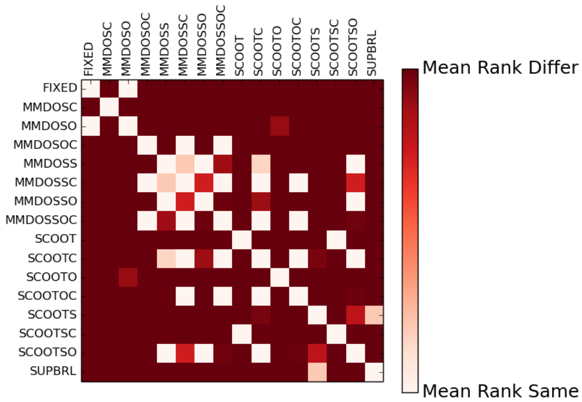
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 7.16: Visual representation of two-sample Mann-Whitney test conducted on ANS (Portland map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

7.3.6 Cumulative Average Number of Stops (CANS)

In *unstructured* traffic on the Portland map, SCOOT has the lowest CANS, see Figure 7.17a. That is, all of the MMDOS variants have higher CANS than SCOOT in the *unstructured* traffic scenario, on the Portland map. Prior to and after the *unstructured* traffic disruption, all of MMDOS variants, excluding MMDOS(OC) and MMDOS(C), perform similar to SUPRL and FIXED. However, during the *unstructured* traffic disruption, MMDOS(S) and MMDOS(SO) have a slight advantage over SUPRL and FIXED. Additionally, MMDOS(OC) and MMDOS(C) have the highest CANS in comparison to the other MMDOS variants and during the disruption both mechanisms have a sharp increase in CANS that does not continue until the disruption terminates.

The CANS in the *football* scenario has some similarities with the *unstructured* traffic scenario, see Figure 7.17c. In both scenarios, MMDOS(OC) and MMDOS(C) have higher CANS than the other MMDOS variants during large portions of the simulation. During the first disruption, all the mechanisms have a similar rate of increase in CANS. However, MMDOS(C) and MMDOS(OC) have a higher peak CANS during the first disruption than the other MMDOS variants. During the second disruption, all the mechanisms, except MMDOS(OC), have lower CANS than SCOOT.

In *directional* traffic, all of the MMDOS variants have lower CANS than SCOOT, see Figure 7.17e. In addition, MMDOS(S), MMDOS(SO), MMDOS(OC) and MMDOS(C) have lower CANS than FIXED and SUPRL. The CANS results in *structured* traffic for the MMDOS variants resemble the CANS results in *directional* traffic. Figure 7.17b shows that MMDOS(C), MMDOS(SO), MMDOS(OC), and MMDOS(S) have lower CANS than SCOOT and SUPRL in *structured* traffic, however, so does FIXED.

In *regional* traffic, Figure 7.17d, prior to the disruption SCOOT has the lowest CANS. However, during the disruption, MMDOS(SC), MMDOS(OC) and MMDOS consistently have lower CANS than SCOOT and FIXED. In *constant* traffic, MMDOS(OC) and MMDOS(C) have lower CANS than SCOOT, FIXED and SUPRL. Also, MMDOS(S) and MMDOS(SO) have lower CANS than SCOOT, FIXED and SUPRL but sometimes has CANS as high as SUPRL. Lastly, in *constant* traffic, MMDOS(O) performs similar to FIXED, both mechanisms have higher CANS than SUPRL.

In *unstructured* traffic, SCOOT(SC) and SCOOT have the lowest CANS, see Figure 7.18a. Prior to the *unstructured* traffic disruption, SCOOT(SO), SCOOT(C), and SCOOT(OC) have CANS that is as low or lower than SUPRL (during this period FIXED has higher CANS than SUPRL). However, during the disruption, some of the SCOOT variants perform worse than SUPRL, FIXED and SCOOT. SCOOT(C), SCOOT(OC) and SCOOT(O) have higher CANS than all of the benchmarks during the *unstructured* traffic disruption. Lastly, after the disruption terminates the CANS of all the mechanisms have a sharp decline, especially, the CANS of SCOOT(C) and SCOOT(OC). This does not occur in *unstructured* traffic on the Phoenix map.

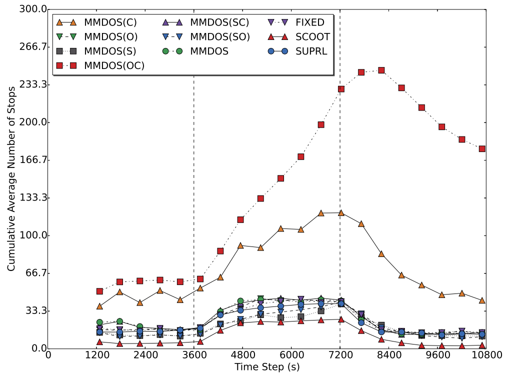
In the *football* scenario, Figure 7.18c, during the first disruption, all of the mechanisms have a similar spike in CANS. During the football match, FIXED has the lowest

CANS but SCOOT(SO) and SCOOT(S) have similar CANS to FIXED. Additionally, during the football match, initially SCOOT(OC) and SCOOT(C) have the highest CANS, however, mid-disruption SCOOT has the highest CANS. SCOOT also has the highest CANS during the second disruption. Also, unlike on the Phoenix map, after the second disruption in the *football* scenario, there is little evidence of recovery after the second disruption terminates on the Portland map.

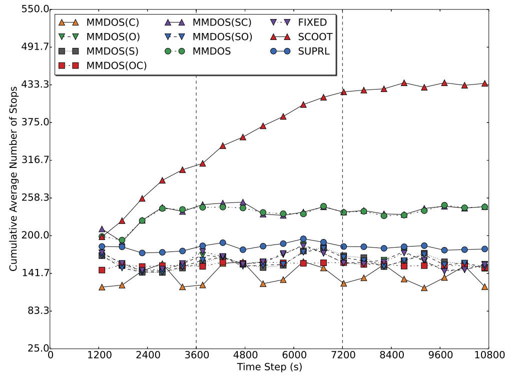
All of the SCOOT variants, except SCOOT(SC), have lower CANS than SCOOT in *directional* traffic, see Figure 7.18e. In *directional* traffic, SCOOT(O), SCOOT(S), and SCOOT(SO) perform in a similar manner to FIXED and SUPRL. Lastly, in *directional* traffic, SCOOT(C) and SCOOT(OC) have lower CANS than SCOOT but not FIXED and SUPRL.

However, in *structured* traffic SCOOT(C) and SCOOT(OC) have the lower CANS, see Figure 7.18b. SCOOT(SC) which performs similar to SCOOT has the highest CANS in *structured* and *constant* traffic. In terms of CANS, SCOOT(C) performs similar to SCOOT in all of the scenarios. Also, in *structured* traffic, SCOOT(SO) and SCOOT(S) performs similar to SUPRL and SCOOT(O) performs similar to FIXED

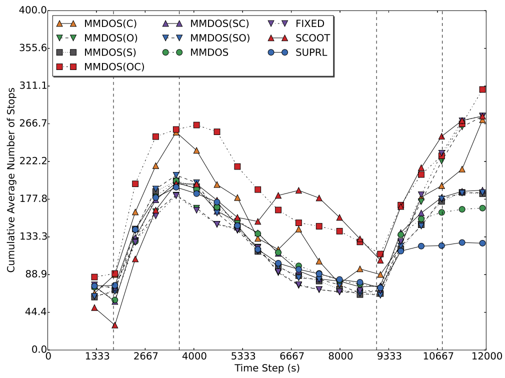
In *regional* traffic, SCOOT(SO) has the lowest CANS, see Figure 7.18d. Also, SCOOT(S) is the only other SCOOT variant that has lower CANS than all three benchmarks in *regional* traffic. Lastly, in *regional* traffic, SCOOT(O) performs similar to FIXED as it does in *structured*, *directional* and *football* traffic. In *constant* traffic, Figure 7.18f, SCOOT(SO) and SCOOT(S) have similar CANS to SUPRL. All three mechanisms have lower CANS than SCOOT and FIXED. Additionally, in *constant* traffic, SCOOT(C) and SCOOT(OC) periodically have the lowest CANS. On the Phoenix map, in *constant* traffic, SCOOT(C), SCOOT(OC) and SCOOT(SC) have the lowest CANS.



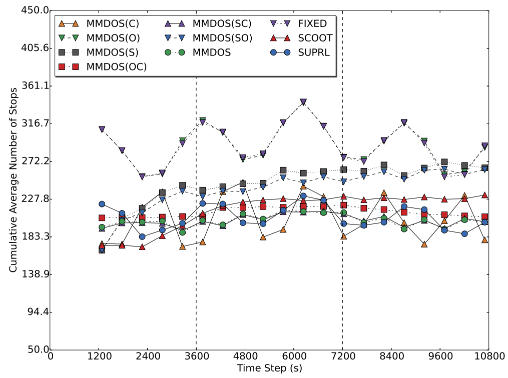
(a) Unstructured.



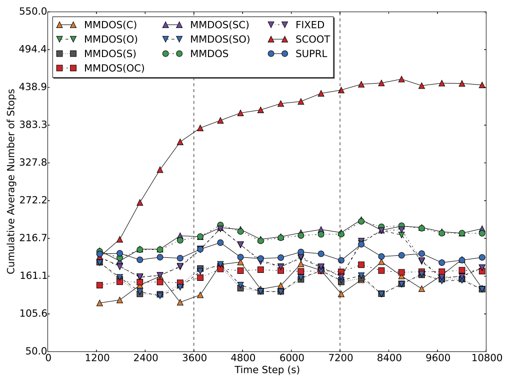
(b) Structured.



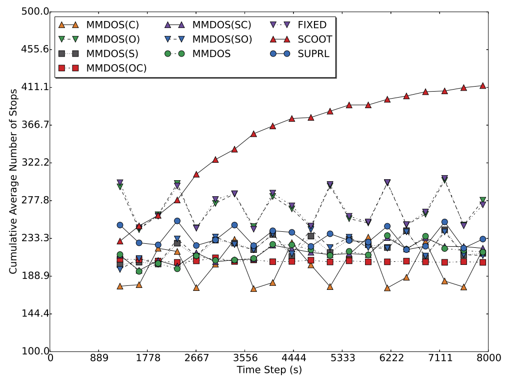
(c) Football



(d) Regional

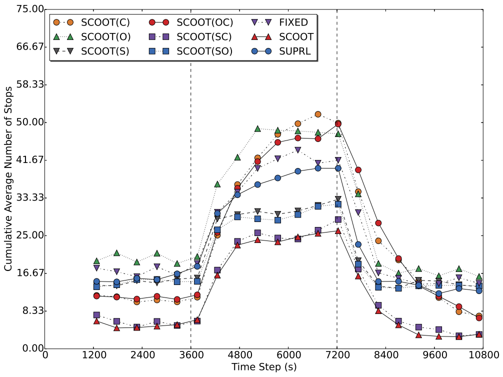


(e) Directional

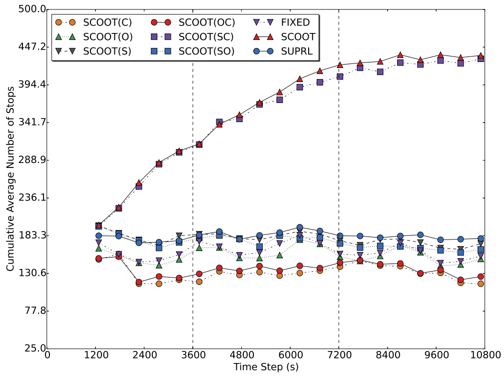


(f) Constant

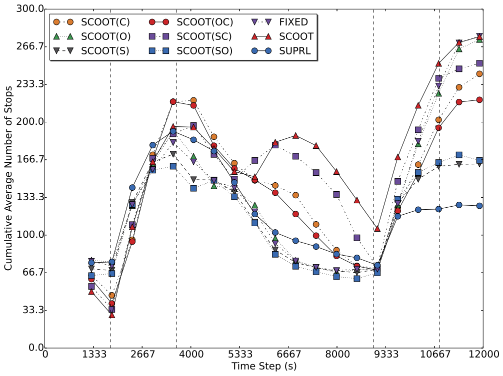
FIGURE 7.17: Cumulative average number of stops (over 30 simulations) on the Portland map. Beginning and ending of disruptions are marked by dotted lines.



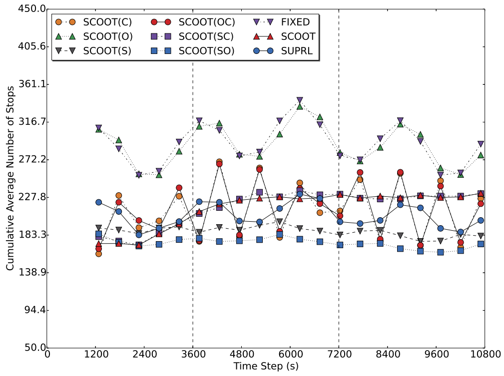
(a) Unstructured.



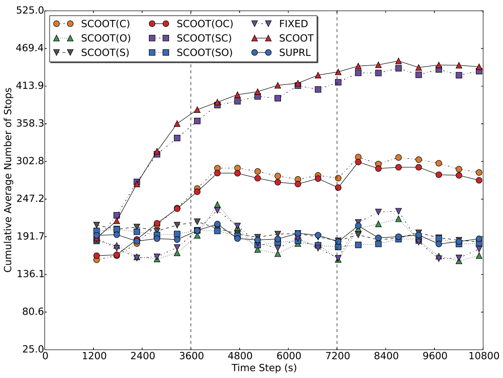
(b) Structured.



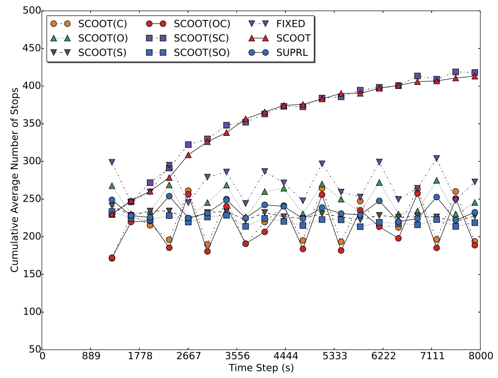
(c) Football



(d) Regional



(e) Directional



(f) Constant

Figure 7.18: Cumulative average number of stops (over 30 simulations) on the Portland map. Beginning and ending of disruptions are marked by dotted lines.

7.4 Summary

This section presents a summary of my findings for the MMDOS and SCOOT variants. The MMDOS and SCOOT variants adjust alternative combinations of *split*, *cycle* and *offset*, i.e., the variants do not adjust all three traffic control parameters at the same time. The ATT, ATD and ANS results discussed in this section are aggregated across both maps.

7.4.1 ATT

Table 7.8 contains the ATT across both maps for the MMDOS and SCOOT variants evaluated in this chapter. Manipulation of alternative combinations of *split*, *cycle* and *offset* improves the ATT performance of MMDOS and SCOOT. In every traffic scenario there is a variant of MMDOS and SCOOT which outperforms the original version, i.e., MMDOS and SCOOT. In only three instances are the ATT of the variant not significant from the original version: the ATT of SCOOT(SO) in *unstructured*, the ATT of SCOOT(SO) in *regional* traffic and the ATT of MMDOS(C) in the *football* scenario, see Figure 7.19.

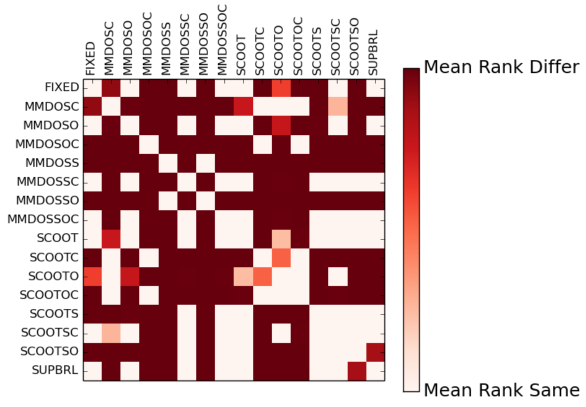
In every traffic scenario, excluding *football* and *regional*, the MMDOS variant with the lowest ATT (MMDOS(C)) also outperforms FIXED, SCOOT and SUPRL. In the *football* traffic scenario, MMDOS(C) only outperforms FIXED and SCOOT; and in the *regional* traffic scenario, MMDOS(C) outperforms FIXED and SUPRL. Additionally, in every scenario, the SCOOT variant with the lowest ATT, outperforms FIXED, SCOOT and SUPRL.

In *football*, *directional*, *structured* and *regional* traffic, MMDOS(C) has the lowest ATT amongst the MMDOS variants. Additionally, in *unstructured* and *constant* traffic, MMDOS(SO) and MMDOS(OC) has the lowest ATT amongst the MMDOS variants. SCOOT(SO) has the lowest ATT amongst the SCOOT variants in *structured*, *directional*, and *regional* traffic. Also, amongst the SCOOT variants, SCOOT(S), SCOOT(C) and SCOOT(OC) have the lowest ATT in *football*, *structured* and *constant* traffic, respectively.

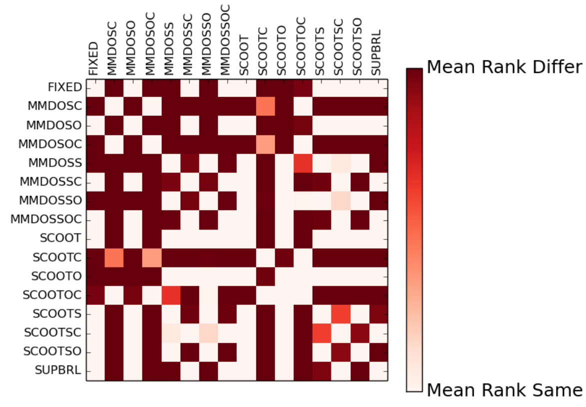
In some of the traffic scenario, certain combinations of traffic control parameter perform well in both families. In *unstructured* traffic, a combination of *split* and *offset* produces the lowest ATT in both families of variants; In *structured* traffic, adjusting *cycle* only produces the lowest ATT in both families of variants; and lastly, in *constant* traffic, a combination of *offset* and *cycle* produces the lowest ATT in both families of variants. Additionally, some combinations of traffic control parameters appear to perform in a similar manner. In every traffic scenario, *split* only and a combination of *split* and *offset* have similar ATT. The same is true for *split+cycle* and *split+cycle+offset* as well as *cycle* and *cycle+offset*

Average Travel Time (ATT) (*std.*)			
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	197.75 (31.92)	237.7 (54.1)	225.27 (41.02)
MMDOS(C)	**165.07 (36.95)**	**188.22 (52.22)**	179.01 (41.09)
MMDOS(O)	197.69 (31.96)	237.44 (54.32)	225.32 (40.94)
MMDOS(OC)	169.13 (41.01)	191.67 (58.14)	**178.96 (44.88)**
MMDOS(S)	187.26 (39.99)	202.27 (51.82)	214 (26.73)
MMDOS(SC)	212.44 (63.25)	208.69 (65.06)	226.8 (35.96)
MMDOS(SO)	187.12 (40.02)	200.28 (51.23)	215.29 (25.93)
MMDOS	212.44 (62.86)	208.7 (64.96)	225.02 (35.92)
SCOOT	252.83 (109.24)	183.36 (54.52)	215.81 (72)
SCOOT(C)	**176.79 (48.63)**	191.4 (54.88)	185.24 (46.71)
SCOOT(O)	193.02 (33.15)	237.62 (53.25)	224.96 (40)
SCOOT(OC)	177.14 (51.39)	191.82 (55.39)	**183.51 (44.64)**
SCOOT(S)	193.84 (48.16)	179.72 (52.26)	210.54 (38.26)
SCOOT(SC)	253.01 (105.74)	181.81 (57.45)	215.23 (74.65)
SCOOT(SO)	191.72 (47.69)	**177.04 (49.07)**	209.78 (36.13)
SUPRL	201.8 (42.7)	195.23 (51.65)	229.62 (24.35)
	Unstructured	*Football*	*Directional*
FIXED	839.34 (296.72)	245.25 (55.66)	208.08 (35.21)
MMDOS(C)	1056.44 (289.12)	**223.93 (81.67)**	**169.6 (38.43)**
MMDOS(O)	813.46 (265.26)	244.69 (54.17)	207.9 (35.23)
MMDOS(OC)	1330.5 (212.55)	244.73 (98.69)	173.09 (43.5)
MMDOS(S)	508.18 (44.23)	225.43 (69.58)	185.41 (36.45)
MMDOS(SC)	647.11 (79.94)	229.32 (76.26)	215.67 (56)
MMDOS(SO)	**508.04 (43.89)**	225.4 (69.25)	185.4 (36.59)
MMDOS	651.03 (88.08)	227.11 (74.75)	215.24 (55.95)
SCOOT	870.98 (446.74)	288.77 (106.27)	258.03 (112.15)
SCOOT(C)	1172.15 (561.75)	295.37 (67.52)	214.05 (83.33)
SCOOT(O)	932.46 (375.04)	246.95 (55.76)	205.58 (36.07)
SCOOT(OC)	1143.19 (561.85)	291.14 (62.68)	212.54 (82.27)
SCOOT(S)	683.03 (166.01)	**205.68 (73.77)**	198.33 (52.83)
SCOOT(SC)	853.29 (366.76)	286.62 (95.82)	254.26 (110.01)
SCOOT(SO)	**656.74 (142.42)**	206.36 (74)	**194.36 (52.98)**
SUPRL	716.68 (150.72)	207.16 (65.13)	204.78 (44.77)

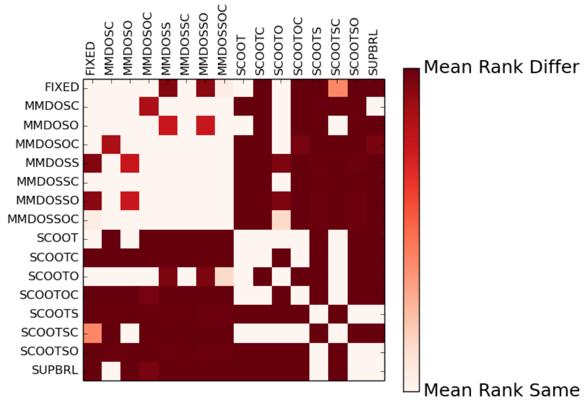
Table 7.8: Average travel times (ATT) for each mechanism and traffic scenario.



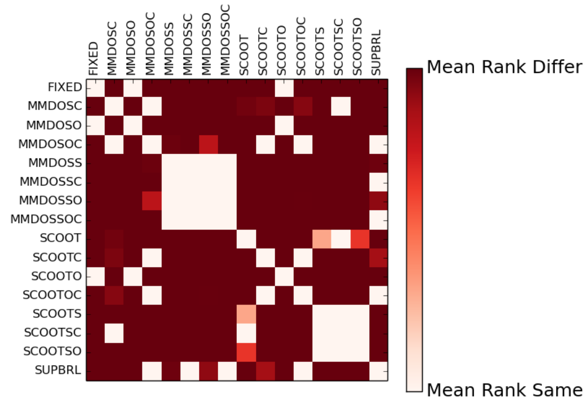
(a) Unstructured.



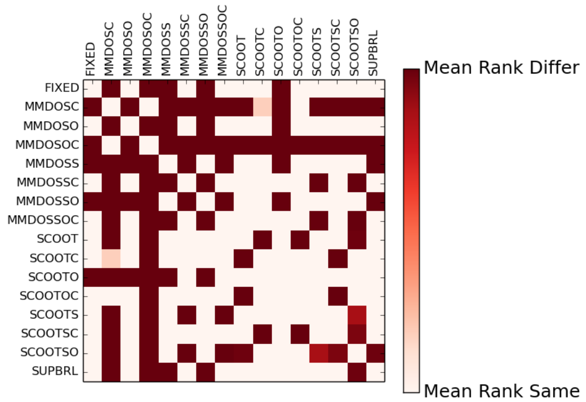
(b) Structured.



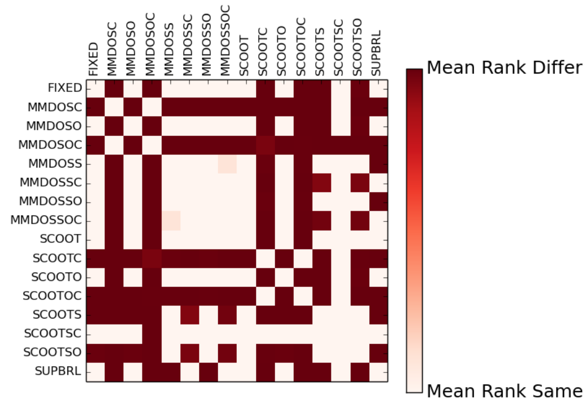
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 7.19: Visual representation of two-sample Mann-Whitney test conducted on ATT results from the 60 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

The ATT results show that adjusting alternative combinations of *split*, *cycle* and *offset* in MMDOS (and SCOOT) will have a significant effect on ATT depending on the traffic condition.

Hypothesis 10 *Adjusting alternative combinations of* split*,* cycle *and* offset *in MMDOS will have a significant effect on* ATT *based on the traffic scenarios.*

> —The ATT of MMDOS(C), MMDOS(OC), MMDOS(S), and MMDOS(SO) is significantly different from the ATT of MMDOS in *unstructured*, *directional* and *structured* traffic. In *regional* traffic, the ATT of MMDOS(C), MMDOS(O), MMDOS(OC) is significantly different from MMDOS and in *constant* traffic, the ATT of MMDOS(C) and MMDOS(OC) is significantly different from MMDOS. Lastly, in *football* traffic, none of the variants performs significantly different from MMDOS.

Hypothesis 13 *Adjusting alternative combinations of* split*,* cycle *and* offset *in SCOOT will have a significant effect on* ATT *based on the traffic scenarios.*

> —The ATT of SCOOT(C) and SCOOT(OC) is significantly different from the ATT of SCOOT in *unstructured*, *structured* and *constant* traffic. In *football* traffic, the ATT of SCOOT(S) and SCOOT(SO) is significantly different from SCOOT; in *directional* traffic, the ATT of SCOOT(C), SCOOT(OC) and SCOOT(SO) is significantly different from SCOOT; and in *constant* traffic, the ATT of SCOOT(C), SCOOT(O) and SCOOT(OC) is significantly different from SCOOT.

7.4.2 ATD

Table 7.9 contains the ATD across both maps for the MMDOS and SCOOT variants evaluated in this chapter. Manipulation of alternative combinations of *split*, *cycle* and *offset* improves the ATD performance of MMDOS and SCOOT. Similar to the ATT of the variants, in every scenario there is a variant of MMDOS and SCOOT which outperforms the original mechanism, in terms of ATD. However, the difference in ATD between the variant and the original mechanism is not significant in all traffic scenarios. The MMDOS variant with the lowest ATD is not significantly different from the ATD of MMDOS in the *football* scenario. Additionally, the SCOOT variant with the lowest ATD is not significantly different from the ATD of SCOOT in *regional* and *unstructured* traffic.

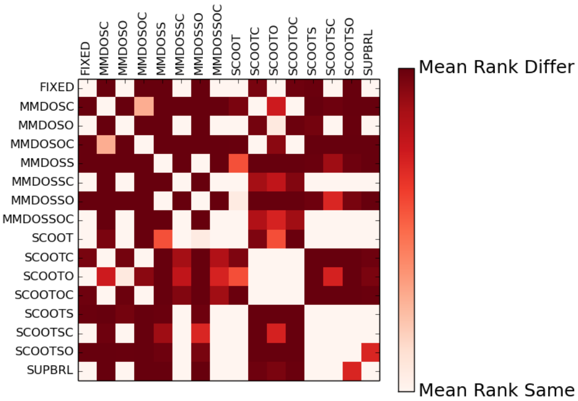
Also, in every traffic scenario, the MMDOS variant with the lowest ATD also outperforms FIXED, SCOOT and SUPRL. However, in two of the traffic scenarios the difference between MMDOS variant with the lowest ATD and the ATD of SCOOT is not significant. In *regional* and *unstructured* traffic, the MMDOS variant with the lowest ATD is not significantly different from SCOOT.

The SCOOT variant with the lowest ATD outperforms FIXED, SCOOT and SUPRL in every traffic scenario as well. However, the SCOOT variant with the lowest ATD is not significantly different from the ATD of FIXED, SCOOT and SUPRL in all traffic scenarios. The SCOOT variant with the lowest ATD is not significantly different from the ATD of FIXED and SUPRL in *unstructured* and *directional* traffic. Lastly, in *regional* traffic, SCOOT variant with the lowest ATD is not significantly different from the ATD of SCOOT.

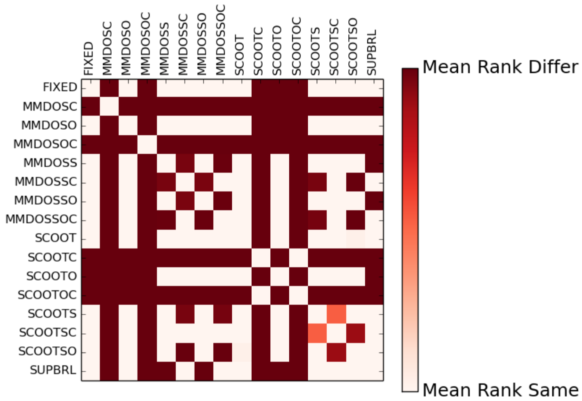
In all of the traffic scenarios, except football, the same combination of traffic control parameters produces the lowest ATD in both MMDOS and SCOOT variants. In *structured*, *regional*, *constant*, and *directional* traffic, MMDOS(OC) produced the lowest ATD and in *unstructured* and *football* traffic, MMDOS(SO) and MMDOS(C) produced the lowest ATD, respectively. Amongst the SCOOT variants, in *structured*, *regional*, *constant*, and *directional* traffic, SCOOT(OC) produced the lowest ATD and in *unstructured* and *football* traffic, SCOOT(SO) has the lowest ATD. In many of traffic scenarios, the mean ATD difference is small, thus, the differences in the performance of the MMDOS and SCOOT variants, in terms of ATD, is modest.

Average Traffic Density (ATD) (*std.*)			
		Traffic Pattern	
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	12.41 (0.2)	18.34 (2.15)	21.06 (0.4)
MMDOS(C)	9.48 (0.4)	13 (0.51)	16.29 (0.48)
MMDOS(O)	12.36 (0.18)	18.24 (2.15)	21.05 (0.43)
MMDOS(OC)	**9.14 (0.4)**	**11.98 (0.23)**	**15.51 (0.21)**
MMDOS(S)	12.34 (1.12)	16 (2.69)	20.97 (2.53)
MMDOS(SC)	13.51 (2.13)	16.57 (3.62)	21.86 (2.38)
MMDOS(SO)	12.32 (1.12)	15.84 (2.55)	21.13 (2.67)
MMDOS	13.44 (2.01)	16.65 (3.64)	21.7 (2.16)
SCOOT	14.66 (3.64)	14.35 (2.68)	20.37 (2.39)
SCOOT(C)	10.06 (0.2)	13.23 (0.59)	16.97 (0.78)
SCOOT(O)	12.12 (0.23)	18.26 (2.02)	20.42 (3.03)
SCOOT(OC)	**10 (0.3)**	**13.11 (0.53)**	**16.8 (0.63)**
SCOOT(S)	12.79 (1.69)	14.21 (2.69)	20.34 (1.53)
SCOOT(SC)	14.67 (3.41)	14.18 (2.9)	20.24 (2.69)
SCOOT(SO)	12.66 (1.69)	14.02 (2.46)	20.35 (1.87)
SUPRL	13.05 (0.92)	15.24 (2.17)	22.52 (3.11)
	Unstructured	*Football*	*Directional*
FIXED	6.61 (3.99)	8.81 (0.82)	13.76 (0.31)
MMDOS(C)	10.77 (8.45)	**7.48 (1.27)**	10.68 (0.28)
MMDOS(O)	6.43 (3.79)	8.83 (0.8)	13.74 (0.31)
MMDOS(OC)	12.87 (9.02)	7.56 (1.18)	**10.13 (0.14)**
MMDOS(S)	3.83 (1.46)	7.71 (0.99)	12.64 (0.7)
MMDOS(SC)	4.96 (2.19)	7.77 (1.17)	14.17 (1.46)
MMDOS(SO)	**3.82 (1.42)**	7.72 (0.99)	12.65 (0.74)
MMDOS	5.33 (3.43)	7.78 (1.18)	14.07 (1.47)
SCOOT	7.11 (5.3)	8.73 (1.3)	15.49 (3.79)
SCOOT(C)	8.32 (5.46)	9.83 (0.83)	13.13 (2.64)
SCOOT(O)	8 (5.89)	8.88 (0.82)	13.53 (0.39)
SCOOT(OC)	9.7 (7.21)	9.79 (0.65)	**13.1 (2.63)**
SCOOT(S)	5.44 (2.98)	7.17 (1.5)	13.48 (1.82)
SCOOT(SC)	6.64 (4.29)	8.76 (1.06)	15.24 (3.62)
SCOOT(SO)	**5.23 (2.78)**	**7.11 (1.4)**	13.21 (1.87)
SUPRL	5.66 (2.96)	7.55 (1.45)	13.8 (0.98)

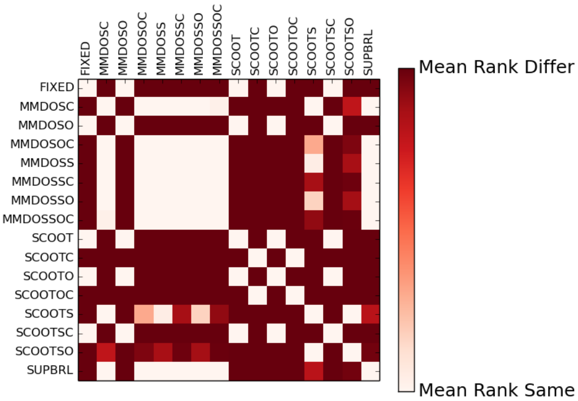
TABLE 7.9: Average traffic density (ATD) for each mechanism and traffic scenario.



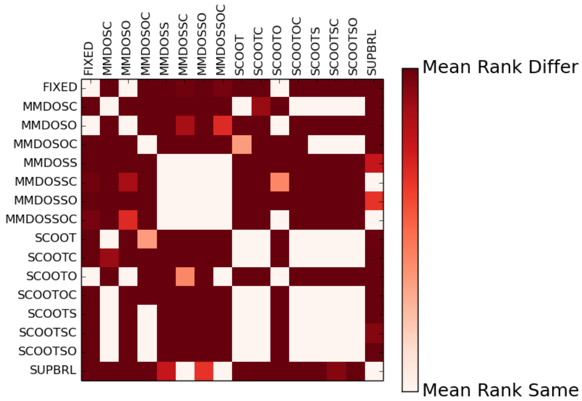
(a) Unstructured.



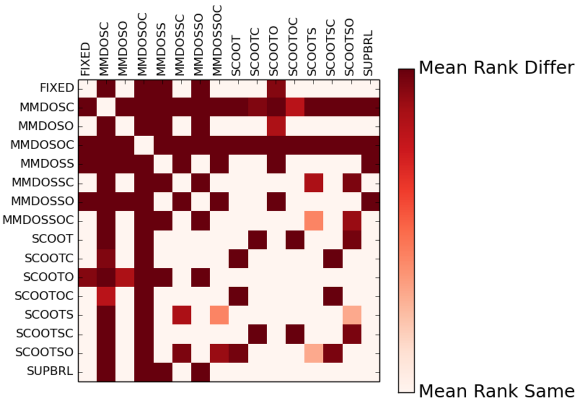
(b) Structured.



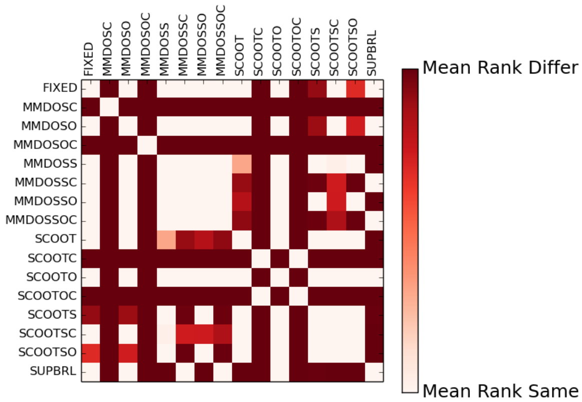
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 7.20: Visual representation of two-sample Mann-Whitney test conducted on ATD results from the 60 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

The ATD results show that adjusting alternative combinations of *split*, *cycle* and *offset* in MMDOS will have a significant effect on ATD depending on the traffic condition.

Hypothesis 11 *Adjusting alternative combinations of* split*,* cycle *and* offset *in MMDOS will have a significant effect on* ATD *based on the traffic scenarios.*

—The ATD of MMDOS(C),MMDOS(OC), MMDOS(S), and MMDOS(SO) is significantly different from MMDOS in *unstructured*, *directional* and *structured* traffic. Also, the ATD of MMDOS(C) and MMDOS(OC) is significantly different from MMDOS in *regional* and *constant* traffic. Lastly, in the *football* scenario, MMDOS(O) is the only variant significantly different from MMDOS.

Hypothesis 14 *Adjusting alternative combinations of* split*,* cycle *and* offset *in SCOOT will have a significant effect on* ATD *based on the traffic scenarios.*

—In *unstructured*, *structured* and *constant* traffic, the ATD of SCOOT(C) and SCOOT(OC) is significantly different from SCOOT; in *football* traffic, the ATD of SCOOT(C), SCOOT(OC), SCOOT(S) and SCOOT(SO) is significantly different from SCOOT; in *directional* traffic, the ATD of SCOOT(C), SCOOT(OC) and SCOOT(SO) is significantly different from SCOOT; and lastly, in *regional* traffic, only SCOOT(O) is significantly different from SCOOT.

7.4.3 ANS

Table 3 contains the ANS across both maps for the MMDOS and SCOOT variants evaluated in this chapter. Manipulation of alternative combinations of *split*, *cycle* and *offset* improves the ANS performance of MMDOS and SCOOT as well as ATT and ATD. Similar to the ATT of the variants, in every scenario there is a variant SCOOT which outperforms the original mechanism, in terms of ANS. Also, excluding the *football* scenario, in every other scenario there is a variant of MMDOS which outperforms the original mechanism. In the *football* scenario, the ANS of the best performing MMDOS is higher than the ANS of MMDOS.

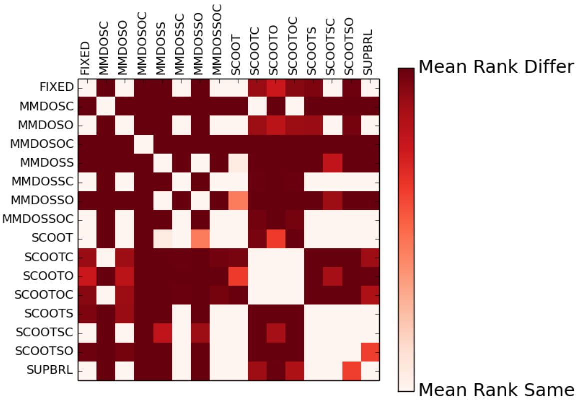
In every traffic scenario, excluding *football*, the MMDOS variant with the lowest ANS also outperforms FIXED, SCOOT and SUPRL. In the *football* traffic scenario, the variant with lowest ANS outperforms SCOOT and FIXED only. The ANS of the best performing MMDOS variant is significantly different from FIXED, SCOOT and SUPRL in *directional*, *structured* and *constant* traffic. The ANS of the best performing MMDOS is not significantly different from SCOOT in *unstructured* and *regional* traffic and SUPRL in *football* and *regional* traffic. Also, in every scenario, the SCOOT variant with the lowest ANS outperforms FIXED, SCOOT and SUPRL.

The ANS of the best performing SCOOT variant is significantly different from FIXED, SCOOT and SUPRL in *structured*, *regional* and *constant* traffic. The ANS of the best performing SCOOT is not significantly different from SCOOT in *unstructured*, SUPRL in any of the traffic scenarios with *unpredictable* traffic, and FIXED in *directional* traffic.

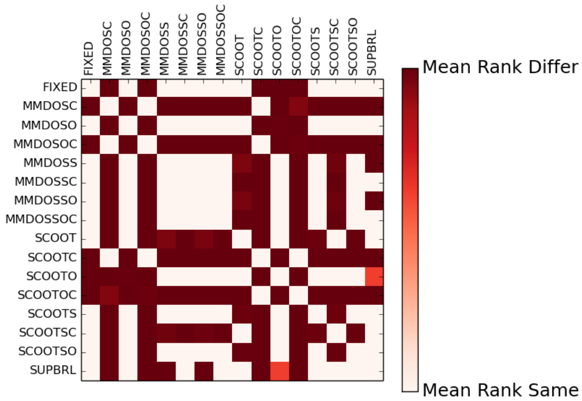
In *directional*, *regional* and *constant* traffic, MMDOS(OC) has the lowest ANS. MMDOS(SO), MMDOS(SC) and MMDOS(C) have the lowest ANS in *unstructured*, *football* and *structured* traffic, respectively. SCOOT(SO) has the lowest ANS in *unstructured*, *football* and *directional* traffic. In *structured* and *constant* traffic, SCOOT(OC) has the lowest ANS.

Average Number of Stops (ANS) (*std.*)			
		Traffic Pattern	
Mechanism	*Structured*	*Regional*	*Constant*
FIXED	115.68 (45.88)	189.45 (91.85)	205.33 (75.29)
MMDOS(C)	**86.28 (36.85)**	127.45 (60.34)	154.38 (63.35)
MMDOS(O)	115.3 (45.79)	188.38 (91.43)	205.1 (74.85)
MMDOS(OC)	87.57 (38.25)	**122.28 (57.95)**	**154.3 (68.48)**
MMDOS(S)	114.3 (56.33)	157.08 (83.22)	190.73 (41.9)
MMDOS(SC)	139.82 (87.25)	126.42 (77.34)	183.53 (44.83)
MMDOS(SO)	114.08 (56.3)	153.48 (80.5)	192.65 (40.67)
MMDOS	138.23 (85.4)	126.88 (77.59)	182.3 (45.18)
SCOOT	189.98 (131.28)	128.87 (75.75)	223.12 (126.29)
SCOOT(C)	84.13 (35.01)	130.78 (63.61)	165.33 (73.57)
SCOOT(O)	111.35 (45.74)	187.37 (89.6)	194.88 (74.58)
SCOOT(OC)	**83.98 (37.09)**	129.62 (62.78)	**161.83 (69.6)**
SCOOT(S)	123.48 (67.12)	118.57 (68.55)	183.42 (58.45)
SCOOT(SC)	188.08 (127.46)	128.22 (77.47)	223.35 (129.3)
SCOOT(SO)	120.77 (65.68)	**112.87 (62.84)**	181.47 (54.22)
SUPRL	125.98 (59.9)	132.07 (69.65)	203.12 (43.92)
	Unstructured	*Football*	*Directional*
FIXED	48.58 (23.13)	96.83 (47.16)	133.03 (54.56)
MMDOS(C)	125.3 (101.06)	91.02 (57.66)	101.93 (48.09)
MMDOS(O)	46.17 (20.34)	96.37 (46.13)	132.77 (54.55)
MMDOS(OC)	176.07 (106.72)	98.78 (62.82)	**100.62 (47.48)**
MMDOS(S)	21.87 (5.75)	84.28 (46.06)	112.9 (50.49)
MMDOS(SC)	33.03 (4.93)	81.4 (48.22)	137.07 (75.8)
MMDOS(SO)	**21.83 (6.16)**	84.38 (46.02)	113.1 (50.99)
MMDOS	38.18 (36.7)	**80.73 (47.76)**	135.52 (75.16)
SCOOT	54.62 (51.38)	96.13 (58.91)	204.75 (141.49)
SCOOT(C)	73.1 (43.98)	99.65 (47)	147.65 (94.87)
SCOOT(O)	70.32 (59.96)	97.8 (47.36)	129.62 (54.24)
SCOOT(OC)	95.78 (78.28)	97.02 (42.86)	146.35 (93.79)
SCOOT(S)	35.42 (11.1)	73.02 (49.16)	131.75 (72.57)
SCOOT(SC)	45.98 (28.49)	93.62 (53.32)	198.3 (136.28)
SCOOT(SO)	**33.5 (9.32)**	**72.22 (47.97)**	**126.12 (71.3)**
SUPRL	38.12 (10.22)	76.43 (46.83)	131.62 (63.31)

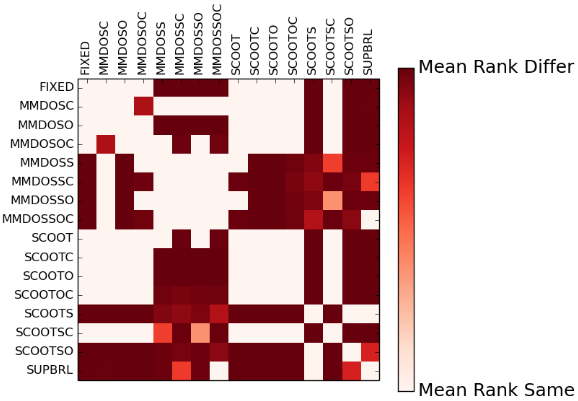
TABLE 7.10: Average *number of stops* (ANS) for each mechanism and traffic scenario.



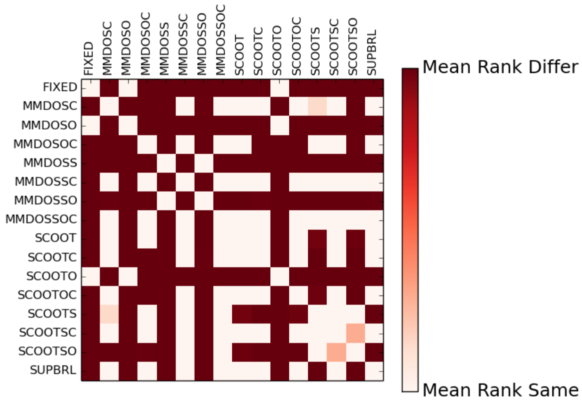
(a) Unstructured.



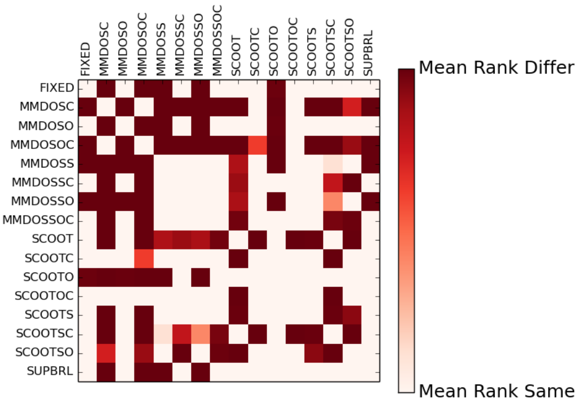
(b) Structured.



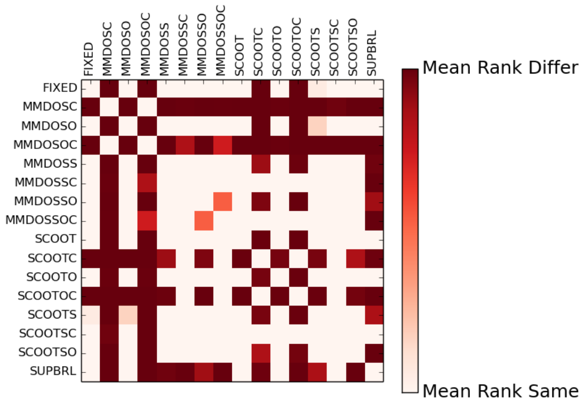
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 7.21: Visual representation of two-sample Mann-Whitney test conducted on ANS results from the 60 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

The ANS results show that adjusting alternative combinations of *split*, *cycle* and *offset* in MMDOS will have a significant effect on ANS depending on the traffic condition.

Hypothesis 12 *Adjusting alternative combinations of* split*,* cycle *and* offset *in MMDOS will have a significant effect on* ANS *based on the traffic scenarios.*

—In *directional* and *structured* traffic, the ANS of MMDOS(C) and MMDOS(OC) is significantly different from MMDOS; in *unstructured* traffic, the ANS of MMDOS(C), MMDOS(OC), MMDOS(S) and MMDOS(SO) is significantly different from MMDOS; in *football* traffic, the ANS of MMDOS(O) and MMDOS(OC) is significantly different from MMDOS; In *regional* traffic, the ANS of MMDOS(O), MMDOS(S), MMDOS(SO) is significantly different from MMDOS; and in *constant* traffic, the ANS of MMDOS(C) is significantly different from MMDOS;

Hypothesis 15 *Adjusting alternative combinations of* split*,* cycle *and* offset *in SCOOT will have a significant effect on* ANS *based on the traffic scenarios.*

—In *unstructured* and *constant*, the ANS of SCOOT(C) and SCOOT(OC) is significantly different from SCOOT; In the *football* scenario, the ANS of SCOOT(S) and SCOOT(SO) is significantly different from SCOOT; in *directional* and *structured* traffic, the ANS of SCOOT(C), SCOOT(OC), SCOOT(S), and SCOOT(SO) is significantly different from SCOOT; in *constant* traffic, the ANS of SCOOT(O), SCOOT(S), and SCOOT(SO) is significantly different from SCOOT.

The performance results show that adjusting alternative combinations of *split*, *cycle* and *offset* in MMDOS and SCOOT will have a significant effect on ATT depending on the traffic scenario. In some of the traffic scenario, certain combinations of traffic control parameter perform well in both families. Results also suggest that *split* or some combination of *split* and another parameter performs better in *unpredictable* traffic than adjusting combinations of parameters that include *cycle*. Similarly, *cycle* combination or *cycle* alone perform better in *predictable* traffic than adjusting combinations of parameters that include *split*.

Chapter 8

Results of Dynamic Coalition Formation Experiments

8.1 Introduction

This chapter presents the traffic simulation results for the dynamic coalition mechanism, DC2. In DC2, intersections form temporary coalitions with neighbouring intersections. This allows an intersection to then adjust its *offset* with its neighbour in order to reduce stops. DC2 also adjusts the *split* along with the *offset* of traffic signals. In this chapter, DC2 is compared to FIXED, SCOOT and SUPRL. FIXED does not adjust any traffic control parameters, that is, the traffic signal timing is static and SUPRL is a Reinforcement-learning based traffic controller which only adjusts the *split*. SCOOT is an adaptive traffic control systems which adjusts all three traffic control parameters, *split*, *cycle* and *offset*. The mechanisms, traffic scenarios and maps presented in this chapter are shown in Table 8.1.

Mechanisms	*Traffic Scenarios*	*Maps*
DC2	Structured	Phoenix
SCOOT	Unstructured	Portland
FIXED	Directional	
SUPRL	Regional	
	Football	
	Constant	

Table 8.1: List of mechanisms, traffic flows and maps presented in this chapter.

This chapter is organised into two major parts, the first presents and analyses the results for each map, Phoenix (Section 8.2) and Portland (Section 8.3), and the second, is analysis across both maps (Section 8.4). Traffic performance is measured using three metrics: *average travel time*, *traffic density*, and *number of stops*. Thus, Section 8.2 presents results from traffic simulations executed on the Phoenix map and is divided into three sub-sections, one for each metric. Likewise, Section 8.3 is divided into three sub-sections, one for each metric, but for traffic, simulations executed on the Portland map. Furthermore, traffic performance is evaluated using six traffic scenarios: *structured*, *unstructured*, *football*, *directional*, *constant*, and *regional*.

Lastly, Section 8.4 contains a summary of the results and addresses the following hypothesis:

Hypothesis 16 *There will be a significant difference in* ATT *of DC2 compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

Hypothesis 17 *There will be a significant difference in* ATD *of DC2 compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

Hypothesis 18 *There will be a significant difference in* ANS *of DC2 compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

8.2 Results: Phoenix

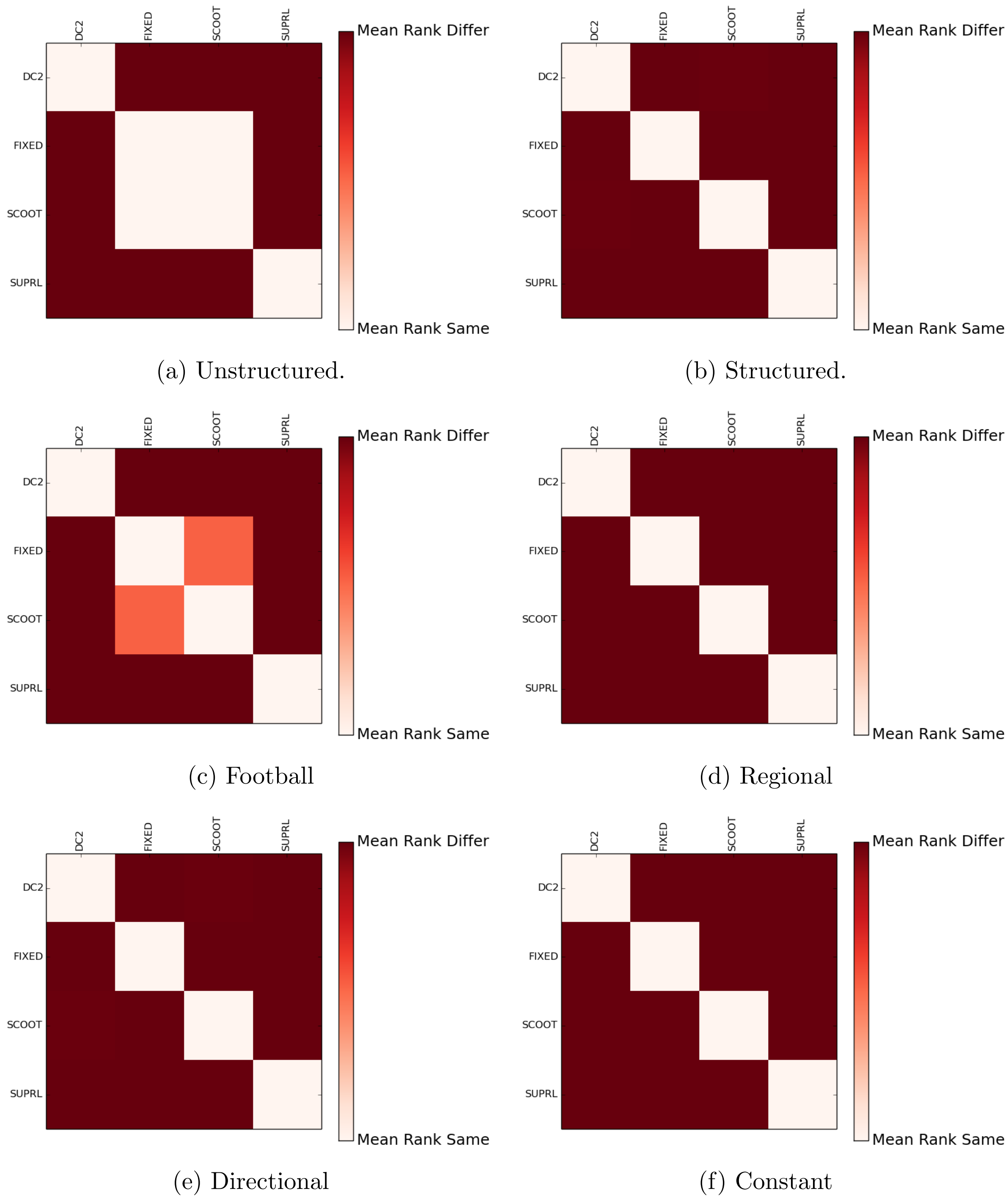
This section presents the results of the experiments executed on the Phoenix map. This section is divided into sub-sections, covering each of the three traffic performance metrics: *average travel time* (Section 8.2.1), *traffic density* (Section 8.2.4), and *number of stops* (Section 8.2.7). The traffic control systems are evaluated in three traffic scenarios with *predictable* traffic flow (*structured*, *regional*, and *constant*) and three traffic scenarios with *unpredictable* traffic flow (*unstructured*, *football*, and *directional*). Results for ATTA (Section 8.2.3) and traffic density on a major artery (Section 8.2.6) are presented for DC2, MMDOS, SCOOT and FIXED in the *unpredictable* traffic scenarios. The Mann-Whitney test is used to determine statistical significance between traffic performance results. The threshold value of $p = .05$ was used to determine whether the null hypotheses (the samples were the same) was rejected. The Mann-Whitney test results are presented in a visual manner in lieu of tables to provide the same information but in a more compact manner than a large table(s).

	Average Travel Time (ATT) (*std.*)		
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
DC2	147.47 (0.78)	150.16 (6.33)	189.19 (4.76)
FIXED	166.11 (1.09)	184.07 (1.32)	184.6 (0.2)
SCOOT	**144.8 (3.44)**	**129.42 (3.71)**	**144.7 (3.52)**
SUPRL	159.48 (1.3)	144.03 (1.42)	206.12 (7.85)
	Unstructured	*Football*	*Directional*
DC2	**515.55 (10.97)**	157.34 (4.6)	149.12 (0.68)
FIXED	1108.81 (168.99)	190.89 (12.8)	173.18 (0.98)
SCOOT	1231.36 (369.63)	184.81 (7.66)	**146.93 (5.16)**
SUPRL	855.66 (78.43)	**142.76 (4.05)**	160.4 (1.26)

Table 8.2: Average travel times (ATT) for each mechanism and traffic scenario.

8.2.1 Average Travel Time (ATT)

In *unstructured* traffic, DC2 has lower ATT than all three benchmarks, Table 8.2. Also, in the *football* scenario, DC2 has lower ATT than FIXED and SCOOT. Lastly, in *directional* traffic, DC2 has lower ATT than FIXED and SUPRL. In all three traffic scenarios with *unpredictable* traffic flow, the ATT performance of DC2 is significantly different from the benchmarks. In *structured* traffic, DC2 has lower ATT than FIXED and SUPRL; DC2 also outperforms SUPRL in the *constant* traffic scenario. However, in *regional* traffic, DC2 only has lower ATT than FIXED. SCOOT outperforms DC2 in all three scenarios with *predictable* traffic compared with the market-based mechanisms with dynamic coalition. In all three scenarios with *predictable* traffic flow, the difference between DC2 and the benchmarks is significant, see Figure 8.1.



(a) Unstructured. (b) Structured.

(c) Football (d) Regional

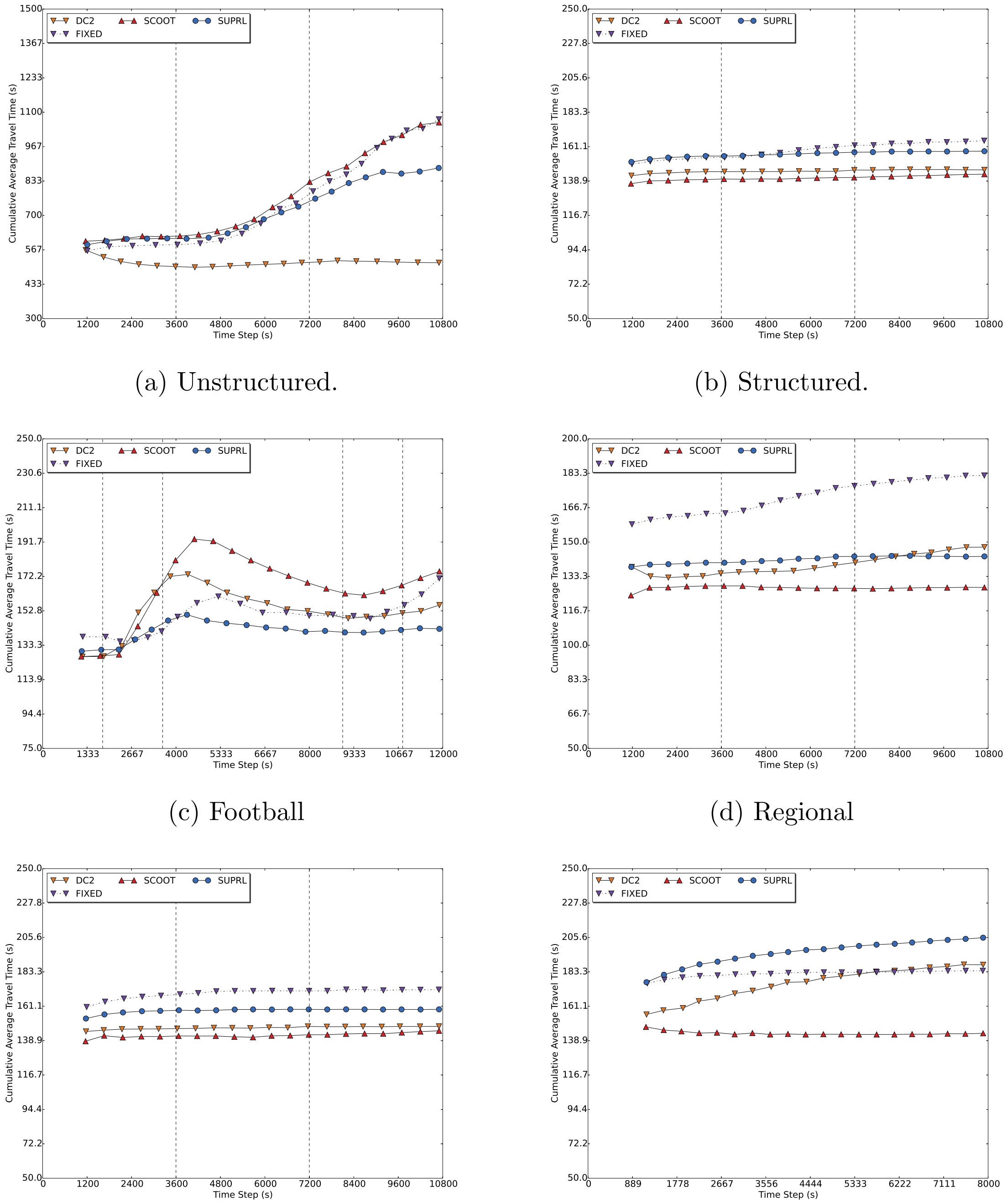
(e) Directional (f) Constant

Figure 8.1: Visual representation of two-sample Mann-Whitney test conducted on ATT (Phoenix map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

8.2.2 Cumulative Average Travel Time (CATT)

On the Phoenix map in the *unstructured* traffic scenario, DC2 has the lowest CATT, see Figure 8.2a. Figure 8.2a shows that in comparison to the other mechanisms, DC2 shows little change in CATT during the *unstructured* traffic disruption. In the *football* scenario, prior to the first disruption, DC2 has the lowest CATT, see Figure 8.2b. However, during the first disruption, DC2 has a sharp increase in CATT but begins to recover sooner than SCOOT during the football match. Figure 8.2b shows that during the football match, the CATT of DC2 decreases and remains lower than SCOOT during the second disruption. After the second disruption, the CATT of DC2 is lower than SCOOT and FIXED but higher than SUPRL. In *directional* and *structured* traffic, DC2 has lower CATT than FIXED and SUPRL but not SCOOT, Figures 8.2e. During the *directional* and *structured* traffic disruption, DC2 displays little change in CATT.

Although DC2 does not have lower CATT than SCOOT in *regional* traffic, prior to and during the *regional* traffic disruption, DC2 has lower CATT than SUPRL. Figure 8.2d shows that after the *regional* traffic disruption, the CATT of DC2 increases and surpasses that of SUPRL. In *regional* traffic, however, DC2 does have lower CATT than SUPRL prior to and during the disruption. In the *constant* traffic scenario, DC2 has lower CATT than SUPRL, see Figure 8.2f. Additionally, the CATT of DC2 reaches the same CATT of FIXED. Lastly, in *constant* traffic, SCOOT has the lowest CATT.



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

(e) Directional

(f) Constant

FIGURE 8.2: Cumulative average travel times (over 30 simulations) on the Phoenix map. Beginning and ending of disruptions are marked by dotted lines.

8.2.3 Average Travel Time on Arrival (ATTA)

Figure 8.3 shows that ATTA in the *unstructured* scenario on the Phoenix map. In comparison to the ATTA of SCOOT and SUPRL, the ATTA of DC2 in *unstructured* traffic displays far less change during and after the *unstructured* traffic disruption, see Figure 8.3. In the *unstructured* traffic scenario, the ATTA of DC2 is more consistent, i.e., the ATTA of DC2 is within a smaller range of values, than with SCOOT & SUPRL. Although DC2 does have an increase in ATTA during the disruption, the vehicles with the highest ATTA under DC2 still have lower travel times than the vehicles with the lowest travel times with SCOOT and SUPRL. Although DC2 has ATTA similar to SCOOT during the both disruptions, Figure 8.4 shows DC2 also has groups of vehicles with ATTA lower than SCOOT during the same period. Additionally, during the football match, DC2 has more vehicles with ATTA less than 100 seconds than SCOOT. However, during the disruptions, the ATTA of DC2 is greater than that of SUPRL. Figures 8.5 shows that in *directional* traffic, the ATTA of DC2 forms two major clusters (This is also true for SUPRL). DC2 has a cluster of vehicles with ATTA less than 100 seconds and a second cluster with ATTA greater than 150 seconds. Additionally, DC2 has many groups of vehicles with higher ATTA than SUPRL and SCOOT during and after the *directional* traffic disruption. However, during this period DC2 also has many groups of vehicles with lower ATTA than SUPRL and SCOOT, see Figure 8.5.

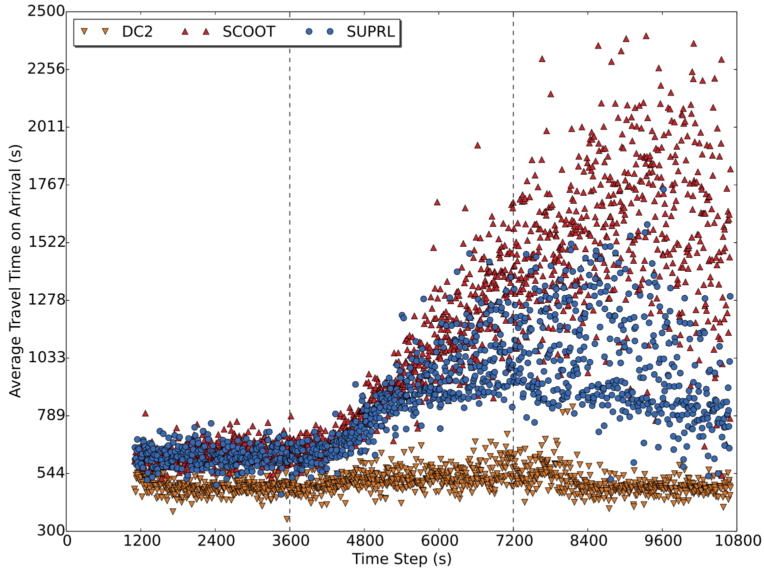


FIGURE 8.3: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *unstructured* traffic on the Phoenix map.

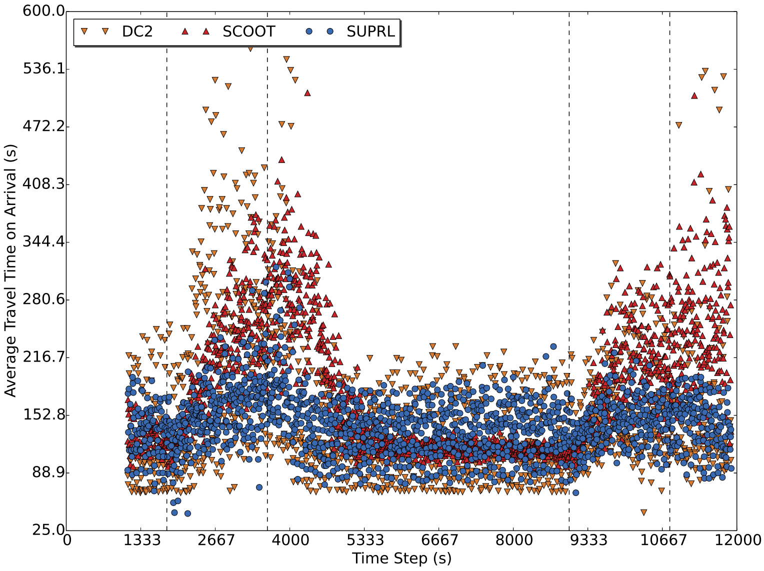


FIGURE 8.4: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *football* traffic on the Phoenix map.

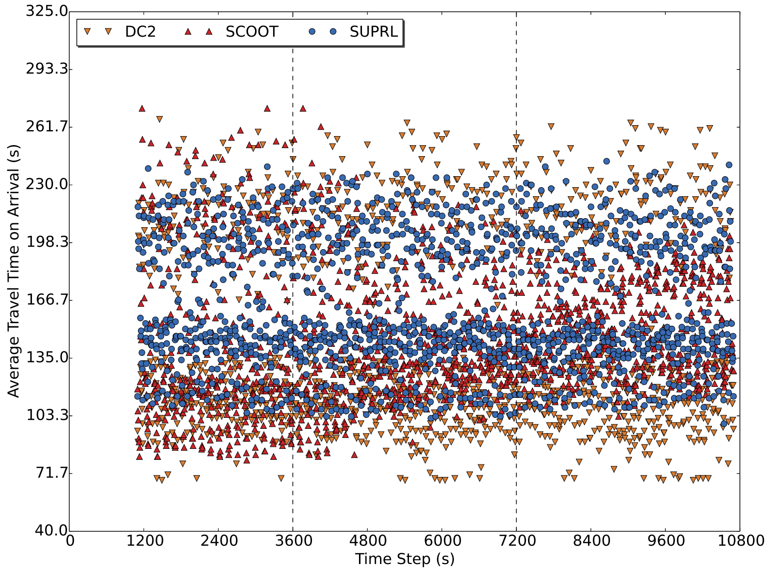


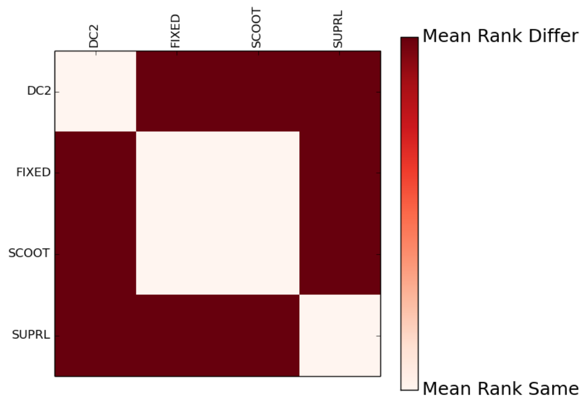
FIGURE 8.5: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *directional* traffic on the Phoenix map.

8.2.4 Density (ATD)

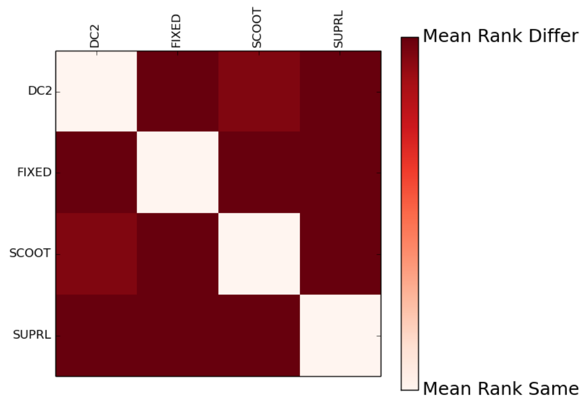
Average Traffic Density (ATD) (*std.*)			
		Traffic Pattern	
Mechanism	*Structured*	*Regional*	*Constant*
DC2	11.23 (0.14)	13.28 (0.55)	23.58 (0.68)
FIXED	12.39 (0.19)	16.22 (0.25)	21.44 (0.09)
SCOOT	**11.07 (0.26)**	**11.72 (0.38)**	**18.07 (0.44)**
SUPRL	12.15 (0.17)	13.1 (0.17)	25.51 (1.01)
	Unstructured	*Football*	*Directional*
DC2	**5.23 (0.2)**	6.76 (0.25)	11.94 (0.11)
FIXED	10.46 (1.31)	8.08 (0.51)	13.5 (0.15)
SCOOT	11.8 (3.42)	7.51 (0.33)	**11.75 (0.39)**
SUPRL	8.53 (0.86)	**6.13 (0.21)**	12.85 (0.2)

TABLE 8.3: Average traffic density (ATD) for each mechanism and traffic scenario.

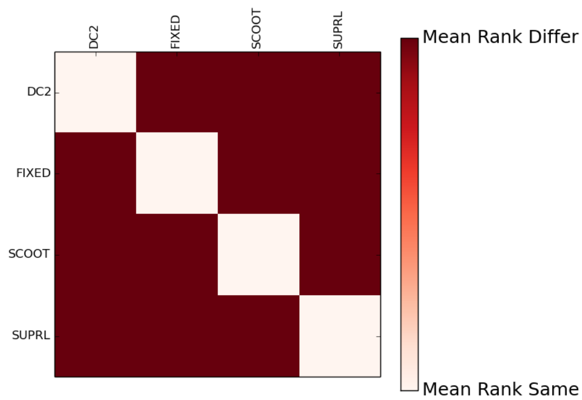
In terms of ATD, DC2 outperforms SCOOT in *unstructured* and *football* traffic, Table 8.3. DC2 also outperforms SUPRL in *unstructured*, *directional*, *structured* and *constant* traffic. Lastly, DC2 outperforms FIXED in all six traffic scenarios. Figure 8.6 shows that the difference between the ATD of DC2 and all three benchmarks is significant except in two traffic scenarios. In *structured* and *regional* traffic, DC2 is not significantly different from SCOOT or SUPRL, respectively.



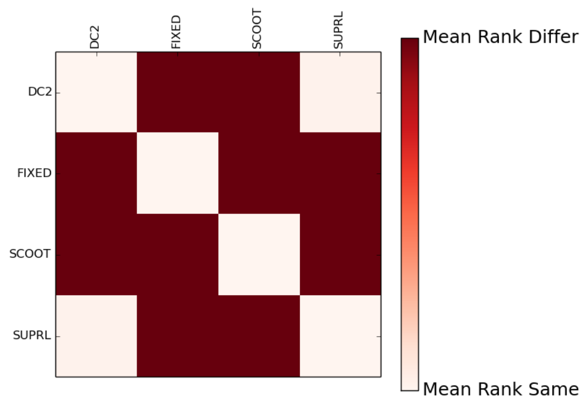
(a) Unstructured.



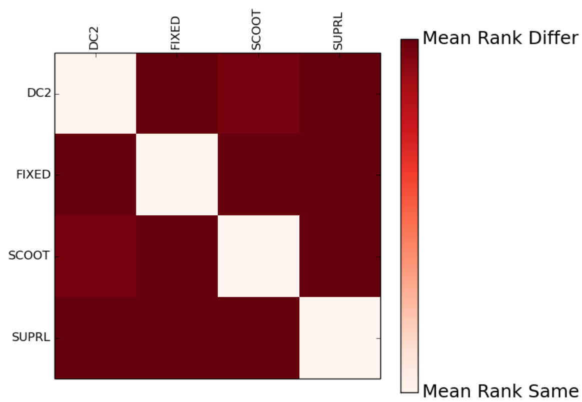
(b) Structured.



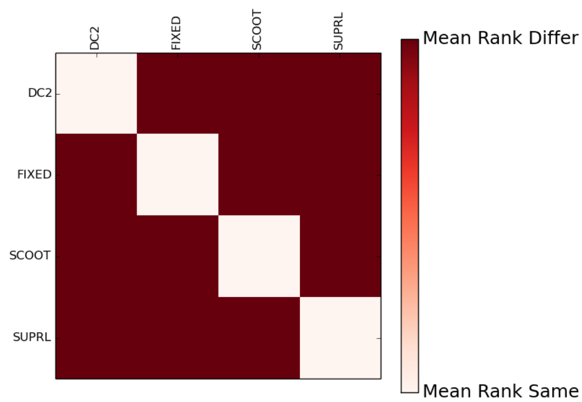
(c) Football



(d) Regional



(e) Directional



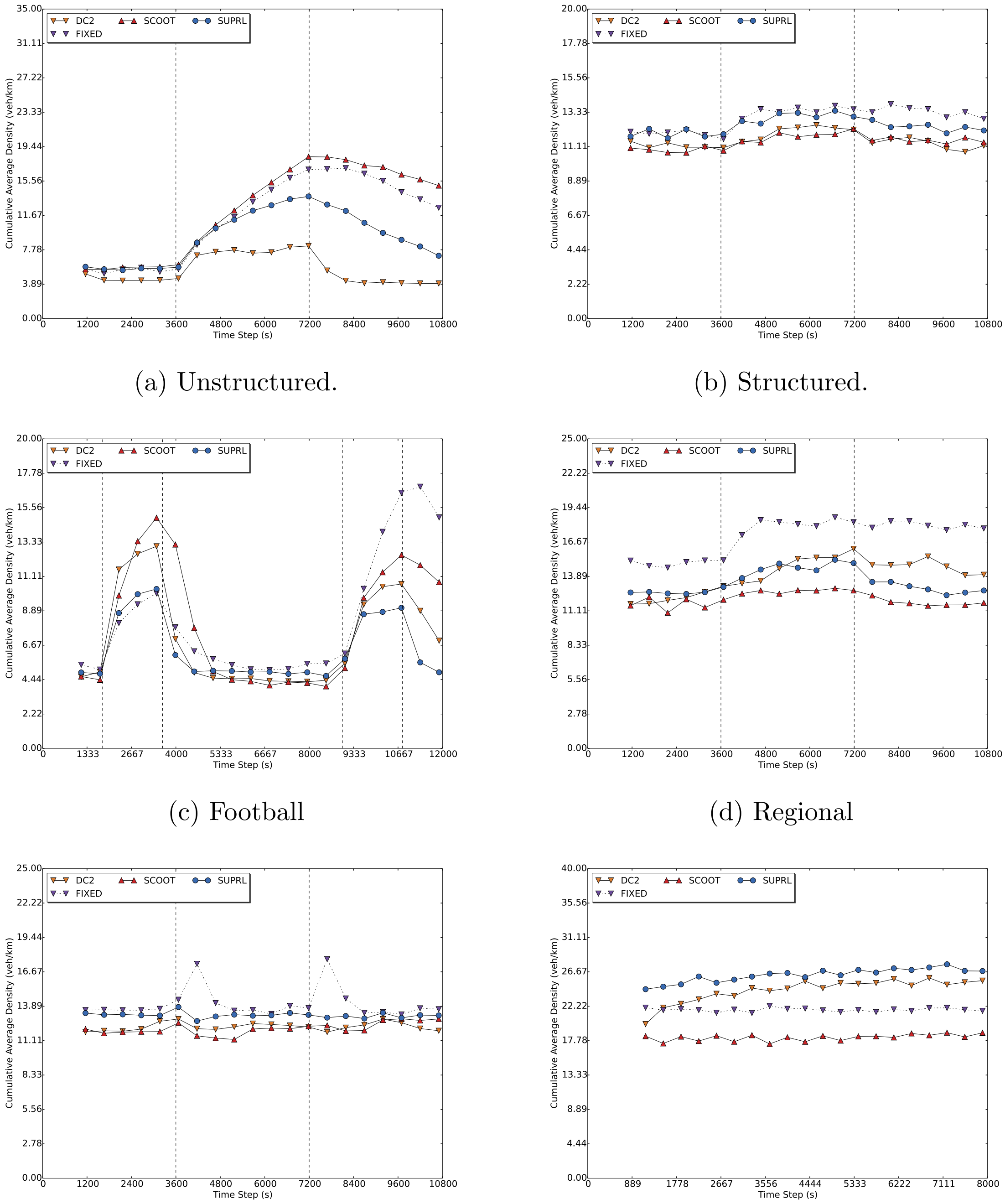
(f) Constant

FIGURE 8.6: Visual representation of two-sample Mann-Whitney test conducted on ATD (Phoenix map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

8.2.5 Cumulative Average Density (CAD)

In *unstructured* traffic, Figure 8.7a, also, DC2 has the lowest CAD, even during the disruption. In the first disruption of the *football* scenario, all the mechanisms display an increase in CAD, however, the CAD of DC2 remain lower than SCOOT's CAD but higher than FIXED and SUPRL, see Figure 8.7c. During the football match the CAD of DC2 recovers much quicker than FIXED and SCOOT. Also, during the football match, DC2 and SCOOT have the lowest CAD. During the second disruption, the peak CAD of DC2 is less than the peak CAD of SCOOT and FIXED (although, SUPRL has the lowest CAD during this period).

In *directional* traffic, Figure 8.7e, DC2 and SCOOT have lower CAD than SUPRL and FIXED. The CAD in *structured* traffic, Figure 8.7b, mirrors the CAD results in *directional* traffic. In *structured* traffic, DC2 and SCOOT have lower CAD than SUPRL and FIXED. At the start of the *regional* traffic scenario, Figure 8.7d, DC2 performs as well as SCOOT and SUPRL. However, during the disruption the CAD of DC2 increases and surpasses that of SCOOT and SUPRL. Lastly, in *constant* traffic, DC2 have higher CAD than FIXED but lower than SUPRL.



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

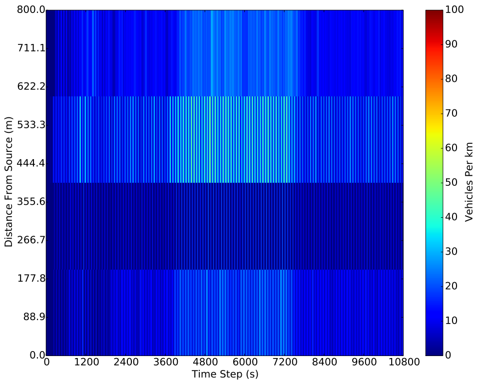
(e) Directional

(f) Constant

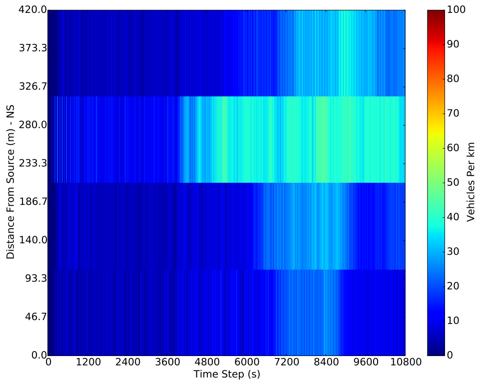
FIGURE 8.7: Cumulative average density (over 30 simulations) on the Phoenix map. Beginning and ending of disruptions are marked by dotted lines.

8.2.6 Density on Major Artery

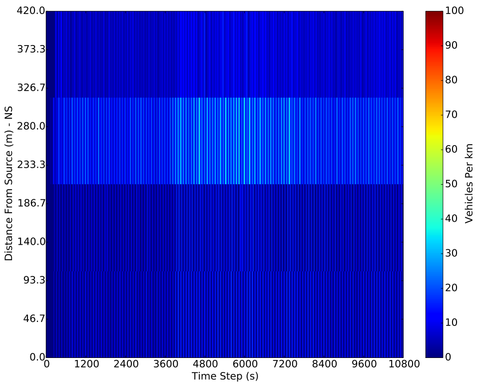
An examination of traffic density on the second artery of *unstructured*, *football* and *directional* traffic, reveals that DC2 and SUPRL have similar traffic density on the second artery. Figure 8.8 shows that in *unstructured* traffic, DC2 and SUPRL have lower traffic density on the second artery compared with SCOOT and FIXED. Again in *football* traffic, the traffic density on the second artery, Figure 8.9, with DC2 and SUPRL are very similar. Traffic density increases on the third road segment during the second with both mechanisms. However, in contrast, SCOOT and FIXED have higher traffic density which spans from the first to the third road segment on the second artery, see Figure 8.9. In *directional* traffic, all of the mechanisms have a decrease in traffic density on the second artery during the *directional* traffic disruption, see Figure 8.10.



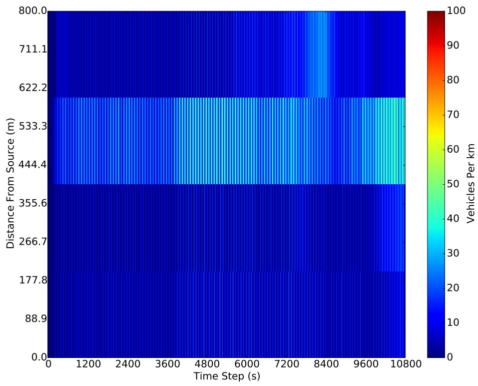
(a) DC2



(b) SCOOT

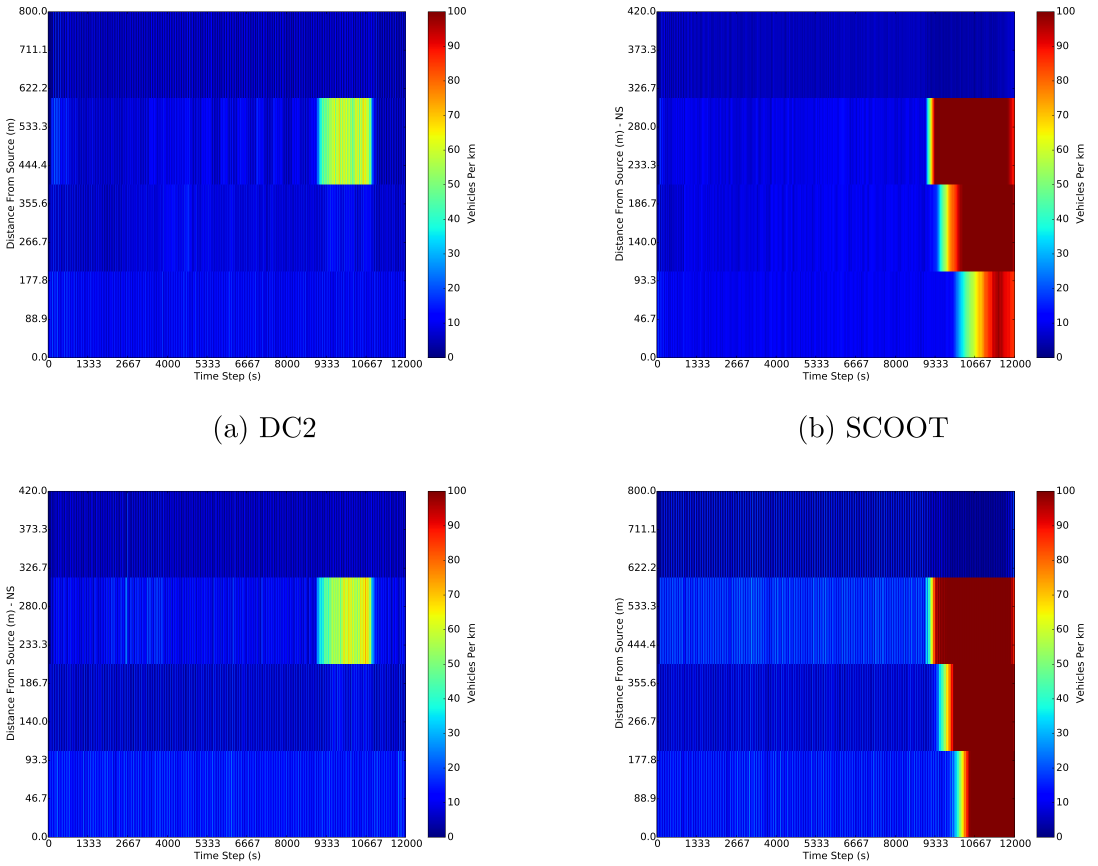


(c) SUPRL



(d) FIXED

FIGURE 8.8: Traffic density on major artery in *unstructured* traffic on Phoenix map.



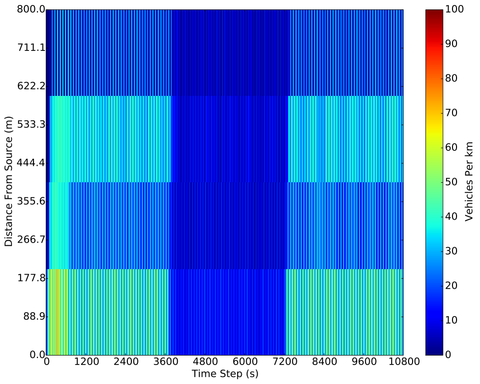
(a) DC2

(b) SCOOT

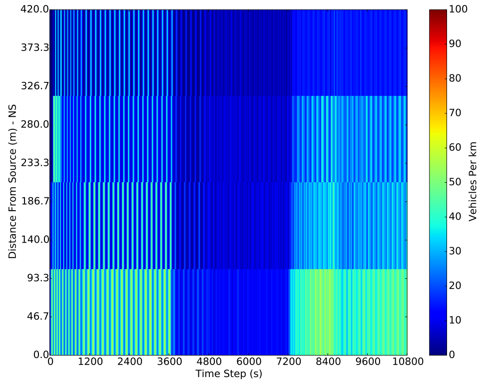
(c) SUPRL

(d) FIXED

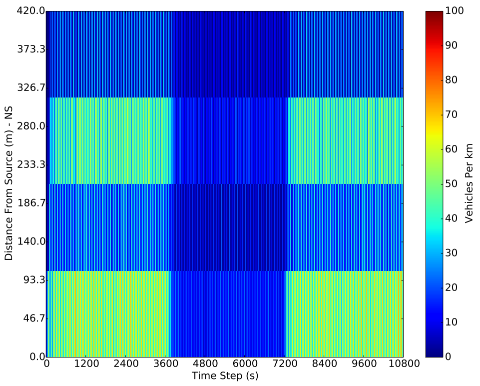
FIGURE 8.9: Traffic density on major artery in *football* traffic on Phoenix map.



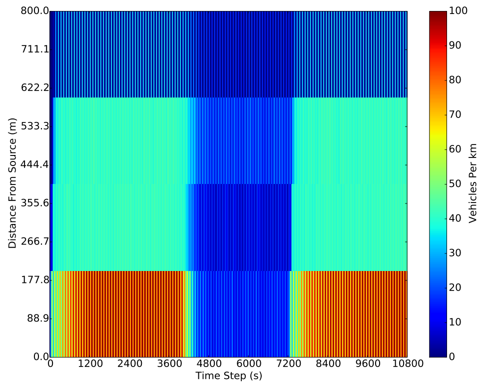
(a) DC2



(b) SCOOT



(c) SUPRL



(d) FIXED

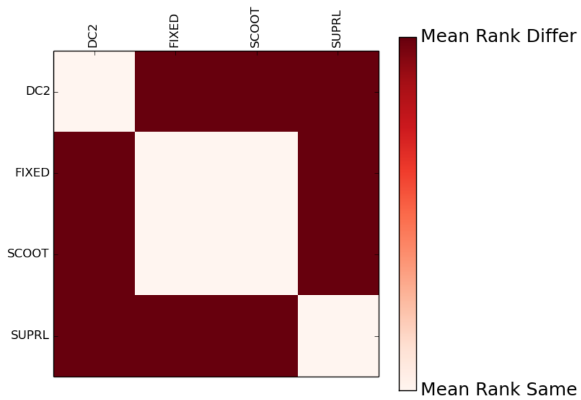
Figure 8.10: Traffic density on major artery in *directional* on Phoenix map.

Average Number of Stops (ANS) (*std.*)			
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
DC2	**58.27 (1.08)**	73.97 (6.39)	151.6 (7.18)
FIXED	70.23 (1.33)	98.4 (1.92)	130.7 (0.6)
SCOOT	60.17 (2.18)	**53.97 (2.92)**	**98.63 (3.6)**
SUPRL	66.63 (1.4)	63.03 (1.33)	160.23 (10.39)
	Unstructured	*Football*	*Directional*
DC2	**19.53 (1.5)**	38.9 (2.54)	**62.63 (0.76)**
FIXED	68.33 (16.6)	50.27 (5.01)	78.97 (1.13)
SCOOT	89.97 (52.73)	38.5 (3.14)	64.57 (3.43)
SUPRL	46.6 (7.83)	**30.13 (1.81)**	68.87 (1.55)

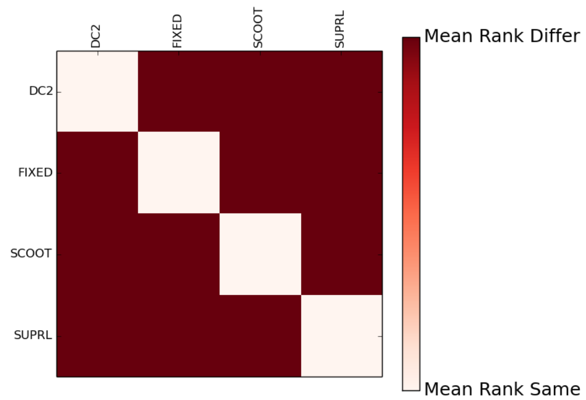
TABLE 8.4: Average *number of stops* (ANS) for each mechanism and traffic scenario.

8.2.7 Vehicle Stops (ANS)

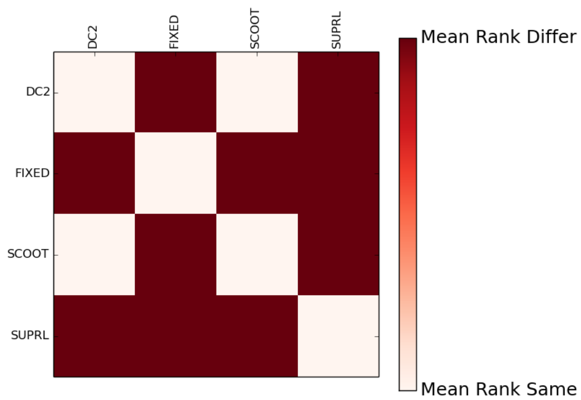
Table 8.4 shows that in *unstructured* and *directional* traffic, DC2 has a lower ANS than all three benchmarks. In the *unstructured* traffic scenario, DC2 halves the ANS compared with SUPRL. The difference in performance between DC2 and the benchmarks in *unstructured* traffic is significant, see Figure 8.11a. However, in *directional* traffic the difference in performance is significant between DC2, FIXED, and SUPRL; the ANS of DC2 is not significantly different from SCOOT. In *football* traffic, DC2 and SCOOT have lower ANS than FIXED but not SUPRL, in addition, the difference in performance between DC2 and SCOOT is not significant, see Figure 8.11c. DC2 also have lower ANS than the benchmarks in *structured* traffic. In *regional* traffic, DC2 has lower ANS than FIXED but not SCOOT and SUPRL. Additionally, in *constant* traffic, DC2 outperforms SUPRL but has higher ANS than SCOOT and FIXED. Figure 8.11 shows that in all three traffic scenarios with *predictable* traffic flow, the ANS of DC2 is significantly different from SCOOT, FIXED and SUPRL.



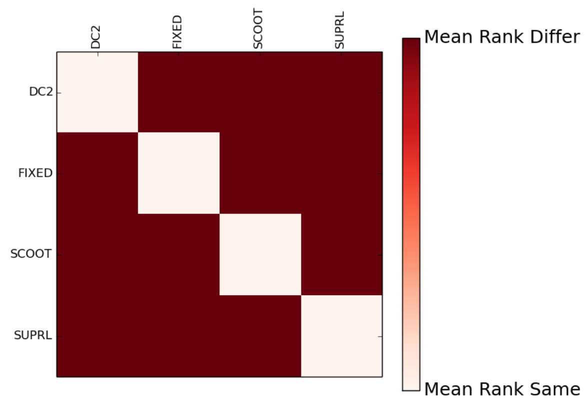
(a) Unstructured.



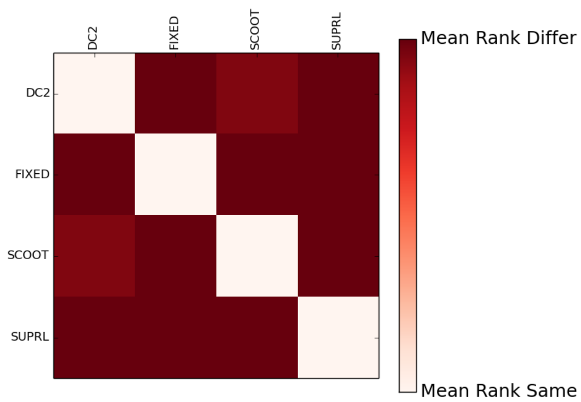
(b) Structured.



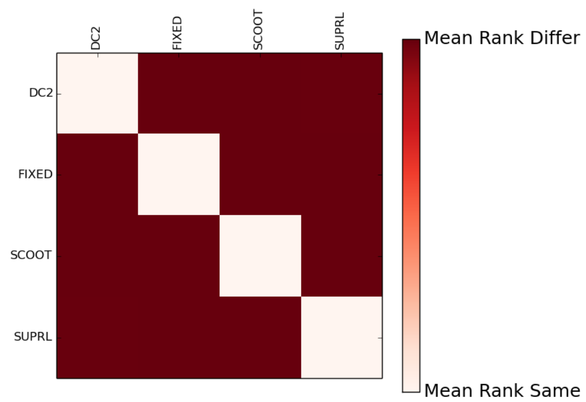
(c) Football



(d) Regional



(e) Directional



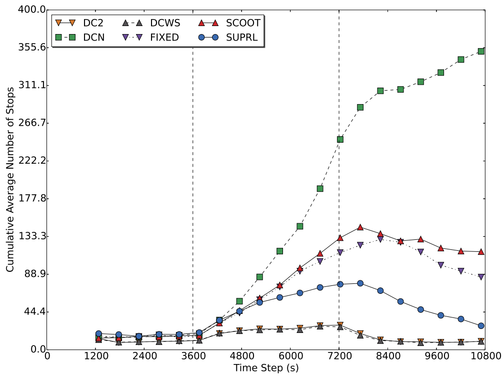
(f) Constant

FIGURE 8.11: Visual representation of two-sample Mann-Whitney test conducted on ANS (Phoenix map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

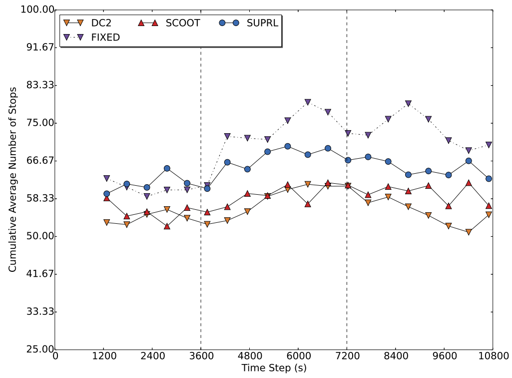
8.2.8 Cumulative Average Number of Stops (CANS)

In the *unstructured* traffic scenario, DC2 has the lowest CANS, Figure 8.12a, throughout the entire scenario. The CANS of DC2 quickly reaches its peak during the disruption, unlike the other mechanisms. In the *football* scenario, Figure 8.12c, during the first disruption DC2 performs similar to SCOOT. However, after the first disruption has ended, DC2 recovers sooner than SCOOT, i.e., DC2 returns to pre-disruption levels of CANS first. During the football match DC2 has lower CANS than SUPRL and FIXED.

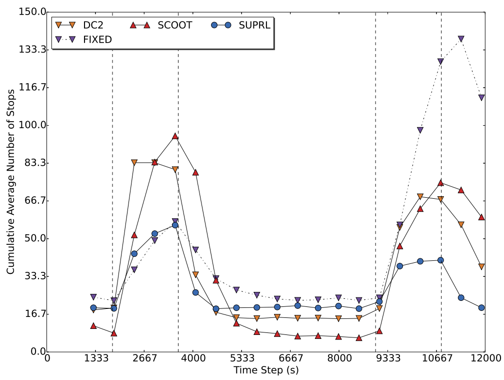
In the *directional* traffic scenario, Figure 8.12e, DC2 performs as well as SCOOT which is better than FIXED and SUPRL. DC2 also performs similar to SCOOT in *structured* traffic, see Figure 8.12b. However, in *regional* traffic, DC2 has higher CANS than SCOOT and SUPRL. Prior to the *regional* traffic disruption, DC2 performs similar to SCOOT and SUPRL, however, once the disruption begins the CANS of DC2 increases until the disruption terminates. In *constant* traffic, DC2 behaves in a similar fashion to SUPRL, see Figure 8.12f. In *constant* traffic, DC2 and SUPRL have higher CANS than FIXED and SCOOT.



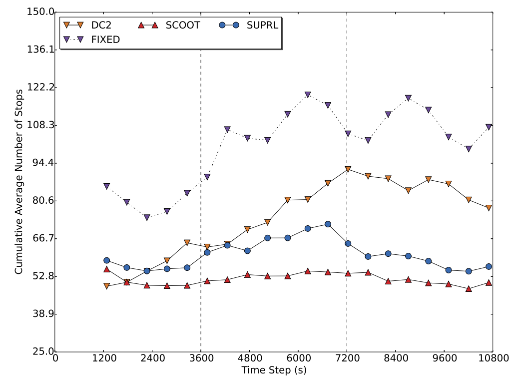
(a) Unstructured.



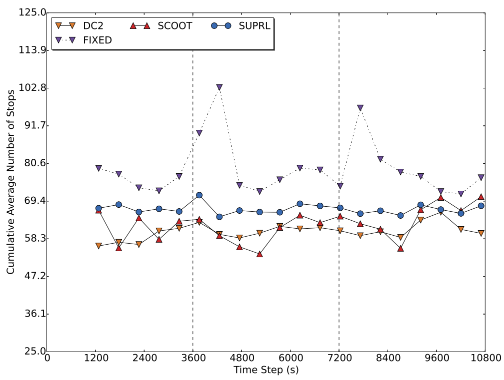
(b) Structured.



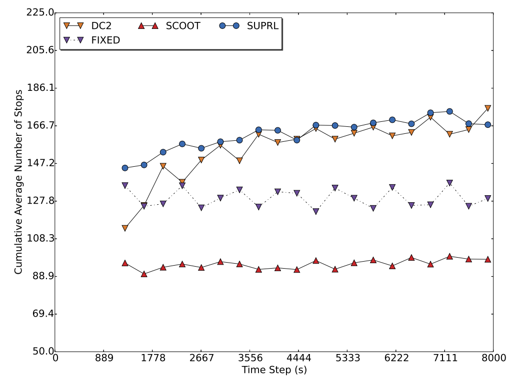
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 8.12: Cumulative average number of stops (over 30 simulations) on the Phoenix map. Beginning and ending of disruptions are marked by dotted lines.

	Average Travel Time (ATT) (*std.*)		
		Traffic Pattern	
Mechanism	*Structured*	*Regional*	*Constant*
DC2	**226.94 (1.3)**	252.08 (3.53)	**240.72 (2.21)**
FIXED	229.38 (0.88)	291.33 (0.66)	265.94 (0.86)
SCOOT	360.86 (11)	**237.29 (3.97)**	286.92 (8.5)
SUPRL	244.13 (1.17)	246.44 (0.94)	253.12 (1.47)
	Unstructured	*Football*	*Directional*
DC2	**483.85 (12.36)**	298.51 (9.42)	**221.39 (0.98)**
FIXED	569.87 (18.05)	299.6 (5.09)	242.97 (0.88)
SCOOT	510.6 (26.52)	392.73 (23.55)	369.13 (4.96)
SUPRL	577.7 (10)	**271.56 (5.62)**	249.16 (1.28)

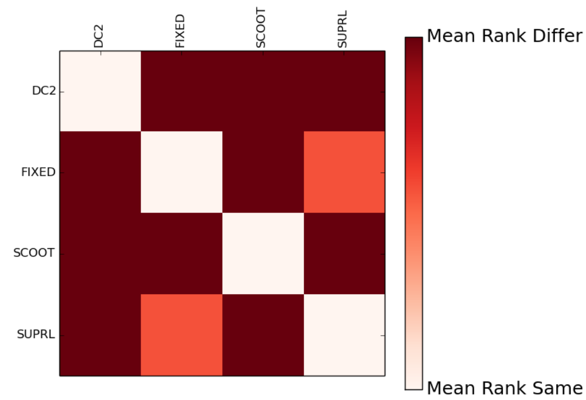
Table 8.5: Average travel times (ATT) for each mechanism and traffic scenario.

8.3 Results: Portland

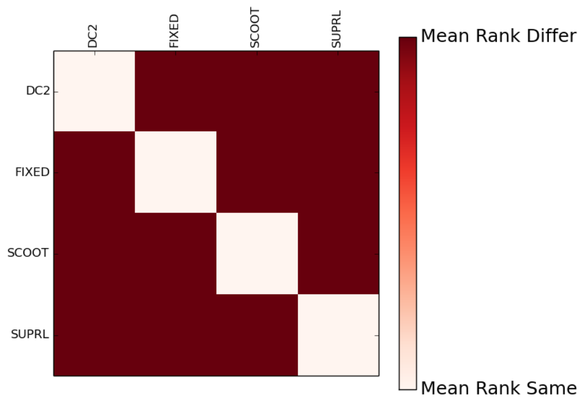
This section presents the results of the experiments executed on the Portland map. This section is divided into sub-sections, covering each of the three traffic performance metrics: *average travel time* (Section 8.3.1), *traffic density* (Section 8.3.4), and *number of stops* (Section 8.3.7). The traffic control systems are evaluated in three traffic scenarios with *predictable* traffic flow (*structured*, *regional*, and *constant*) and three traffic scenarios with *unpredictable* traffic flow (*unstructured*, *football*, and *directional*). Results for ATTA (Section 8.3.3) and traffic density on a major artery (Section 8.3.6) are presented for DC2, MMDOS, SCOOT and FIXED in the *unpredictable* traffic scenarios. The Mann-Whitney test is used to determine statistical significance between traffic performance results. The threshold value of $p = .05$ was used to determine whether the null hypotheses (the samples were the same) was rejected. The Mann-Whitney test results are presented in a visual manner in lieu of tables to provide the same information but in a more compact manner than a large table(s).

8.3.1 Travel Time (ATT)

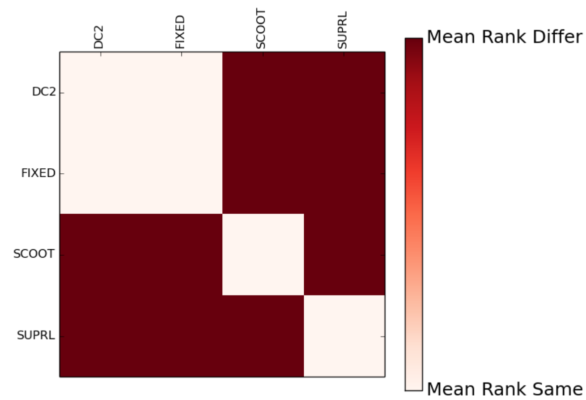
In *unstructured* traffic, DC2 has lower ATT than FIXED, SCOOT and SUPRL (see Table 8.5). Figure 8.13a shows that the difference between DC2 and the benchmarks is significant. In the *football* scenario, DC2 has lower ATT than SCOOT (and slightly lower than FIXED) but DC2 does not outperform SUPRL. The ATT of DC2 is not significantly different from FIXED, Figure 8.13c, in the *football* scenario. In *directional*, *structured* and *constant* traffic, DC2 has lower ATT than FIXED, SCOOT and SUPRL. Additionally, Figure 8.13 shows that in all three scenarios, the difference between the ATT of DC2 and the benchmarks is significant. However, in some scenarios, the magnitude of the difference is small. For example, in *structured* traffic, the difference in ATT between DC2 and FIXED is approximately 2.44 seconds. In *regional* traffic, DC2 has lower ATT than FIXED but higher ATT than SCOOT and SUPRL.



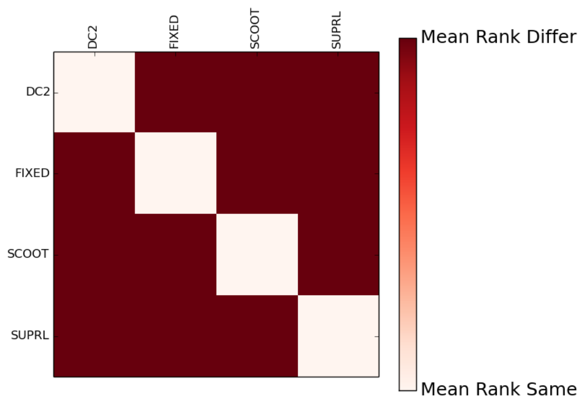
(a) Unstructured.



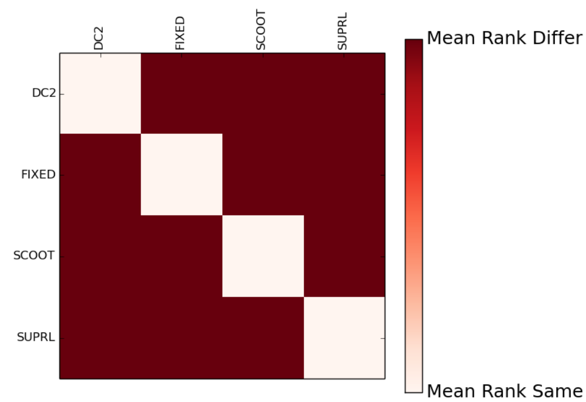
(b) Structured.



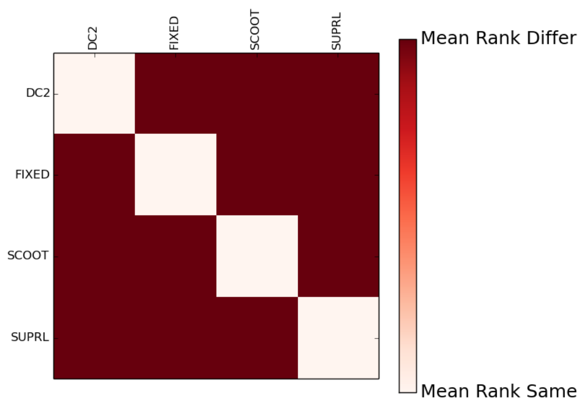
(c) Football



(d) Regional



(e) Directional



(f) Constant

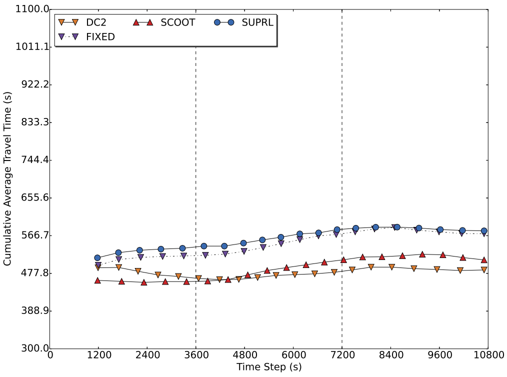
FIGURE 8.13: Visual representation of two-sample Mann-Whitney test conducted on ATT (Portland map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

8.3.2 Cumulative Average Travel Time (CATT)

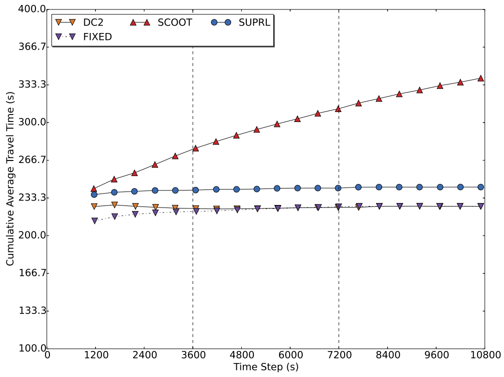
DC2 outperforms all of the benchmarks in two of the traffic scenarios with *unpredictable* traffic. First, in the *unstructured* traffic scenario, initially, the CATT of DC2 is greater than SCOOT, however, during the *unstructured* traffic disruption, DC2 maintain CATT lower than SCOOT, see Figure 8.14a. Figure 8.14a also shows that DC2 has lower CATT than all the other mechanisms after the disruption ends. DC2 performs well in terms of CATT in *unstrctured* traffic on the Phoenix map as well. Second, in the *directional* traffic scenario, DC2 has lower CATT than the benchmarks, see Figure 8.14e. During the *directional* traffic disruption, DC2 shows very little change in cumulative average travel time, similar to the way SUPRL behaves.

In the *football* scenario, DC2 performs as well as SUPRL, see Figure 8.14c. During the first disruption all the mechanisms have a similar rate of increase in CATT. However, during the football match, travel times with SCOOT continue to increase beyond DC2 and SUPRL. The CATT of DC2 and SUPRL plateaus sooner than SCOOT during the football match. Additionally, DC2 and SUPRL show signs of recovery during the match and no adverse effects during the second *football* disruption. However, in the *football* scenario, FIXED has the lowest CATT from beginning of the football match and onward, in comparison to the other mechanisms.

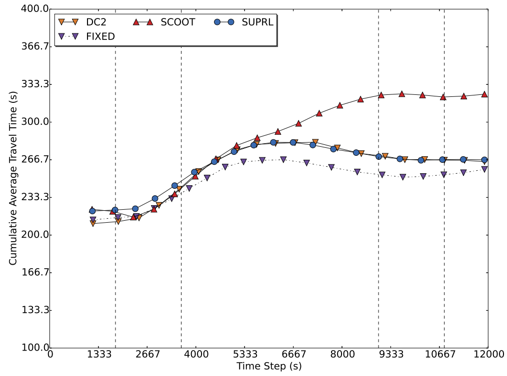
In *structured* and *constant* traffic, DC2 has the lowest CATT, see Figure 8.14. However, in *structured* traffic, the CATT of FIXED is nearly identical to DC2, see Figure 8.14b. In *constant* traffic, DC2 has lower CATT than SCOOT, FIXED and SUPRL. In contrast, in both scenarios DC2 does not have the lowest CATT on the Phoenix map. Finally, in *regional* traffic, Figure 8.14d, DC2 initially has lower CATT than SUPRL (but higher than SCOOT). During the disruption, the CATT of DC2 increases and behaves similar to SUPRL for the remainder of the scenario.



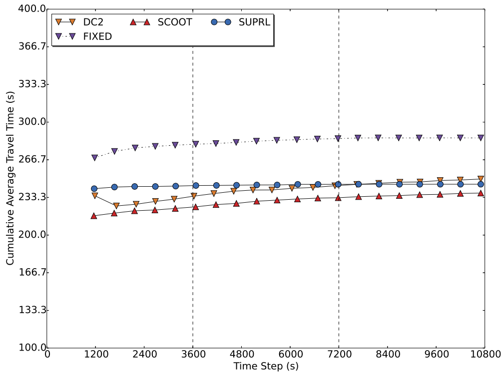
(a) Unstructured.



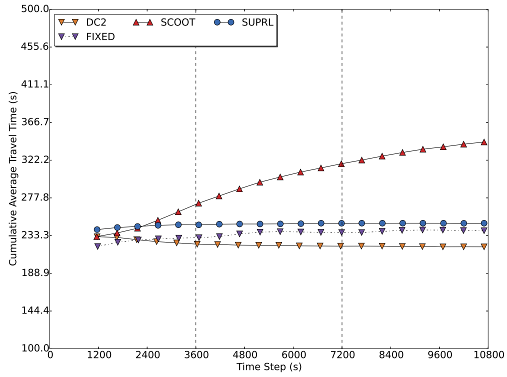
(b) Structured.



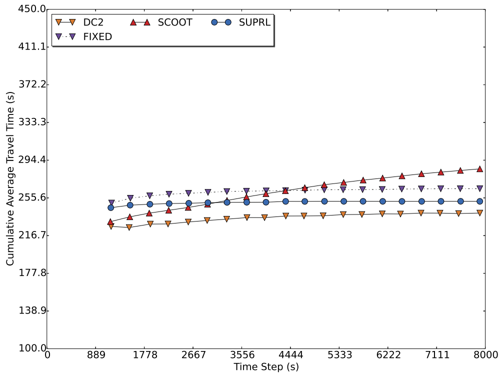
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 8.14: Cumulative average travel times (over 30 simulations) on the Portland map. Beginning and ending of disruptions are marked by dotted lines.

8.3.3 Average Travel Time on Arrival (ATTA)

Figure 8.15 shows that DC2 has similar ATTA to SCOOT in *unstructured* traffic. In general, with *unstructured* traffic, the vehicles with the highest ATTA with DC2 have lower ATTA than SUPRL In the *football* scenario, DC2 has a similar range in ATTA to SUPRL, see Figure 8.16. DC2 also has a similar ATTA to SUPRL near the end of the football match, DC2 and SUPRL have lower ATTA than SCOOT during this period of time. Figure 8.16 also shows that during and after the second disruption, DC2 has a small group of vehicles with higher ATTA than both SCOOT and SUPRL. DC2 with *directional* traffic, on the other hand, has many vehicles with lower ATTA than SCOOT and SUPRL, see Figure 8.17. In *directional* traffic, the ATTA of SUPRL is entirely within the upper bound of DC2's ATTA and SCOOT's ATTA is greater than both DC2 and SUPRL.

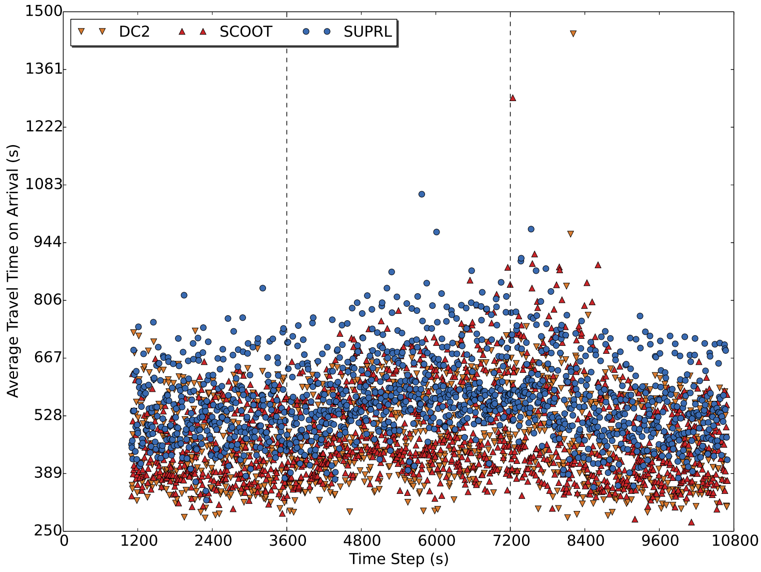


FIGURE 8.15: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *unstructured* traffic on the Portland map.

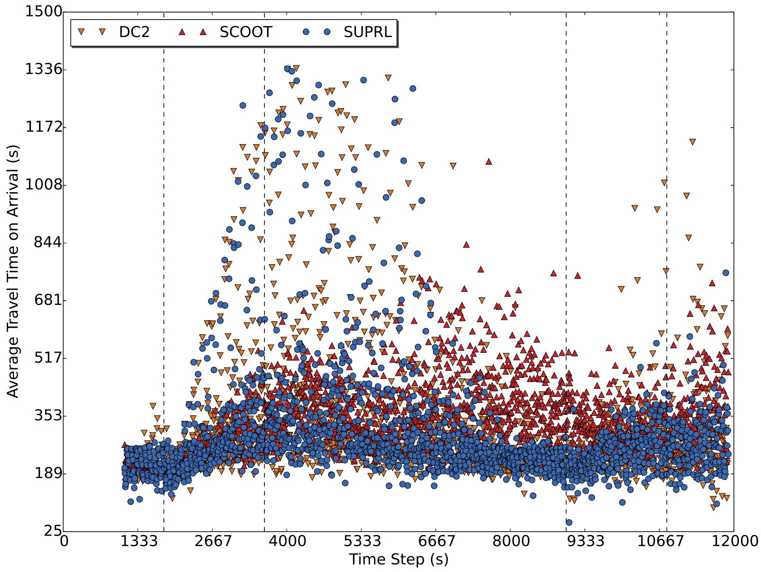


FIGURE 8.16: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *football* traffic on the Portland map.

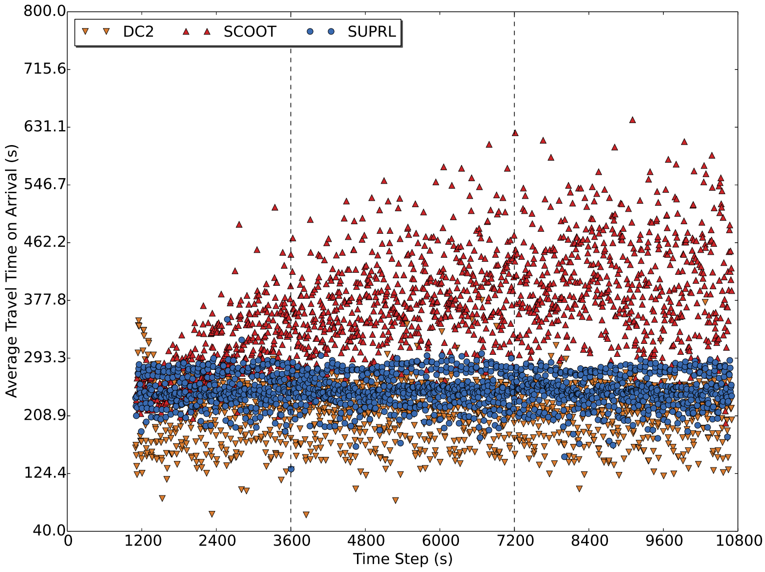


FIGURE 8.17: The average travel times of vehicles that have completed their journey at each time step (over 30 simulations) in *directional* traffic on the Portland map.

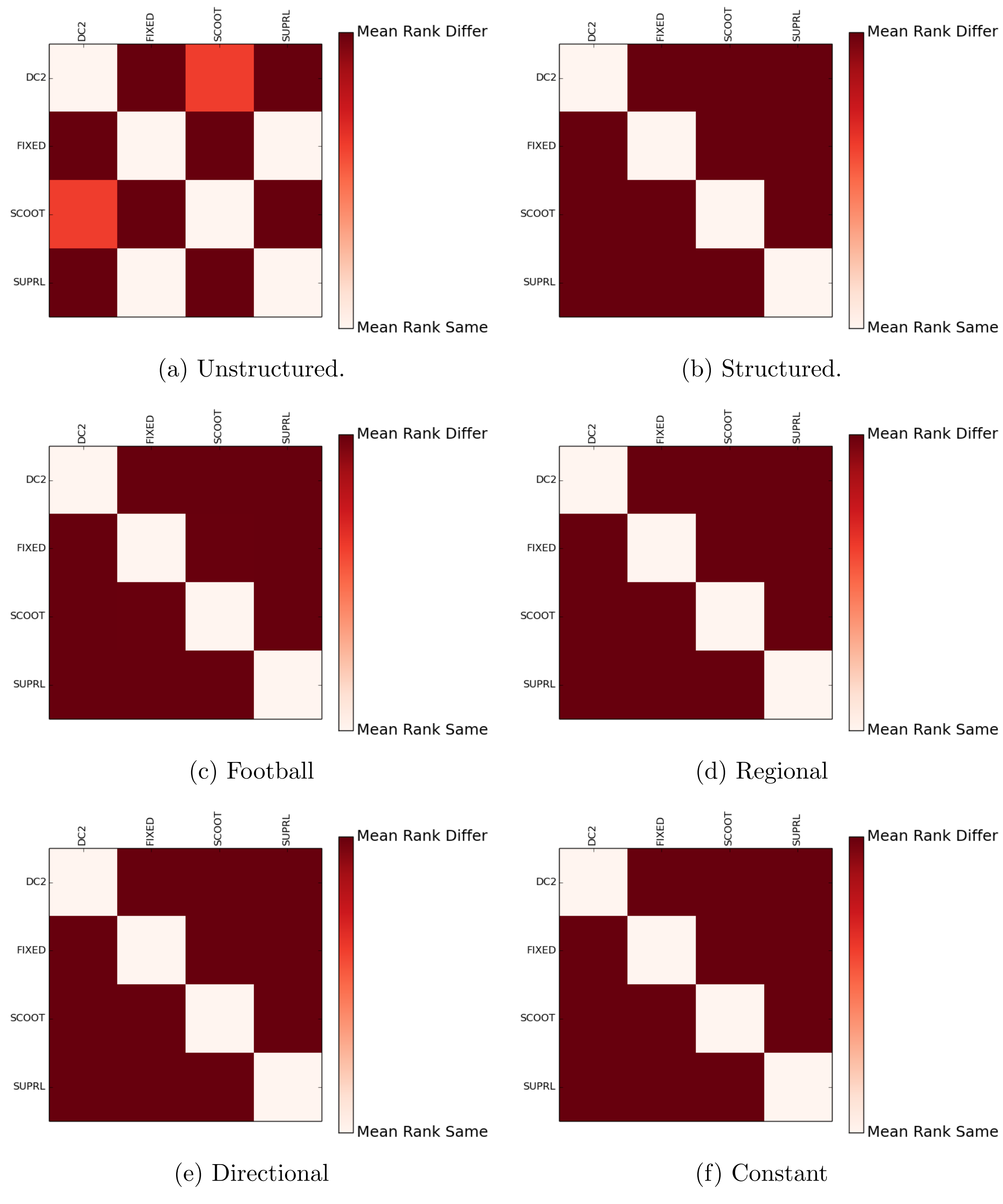
8.3.4 Density (ATD)

Average Traffic Density (ATD) (*std.*)

	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
DC2	13.43 (0.15)	18.4 (0.41)	**18.58 (0.24)**
FIXED	**12.42 (0.21)**	20.46 (0.2)	20.69 (0.16)
SCOOT	18.24 (0.58)	16.98 (0.39)	22.66 (0.75)
SUPRL	13.94 (0.23)	**17.39 (0.2)**	19.52 (0.19)
	Unstructured	*Football*	*Directional*
DC2	**2.35 (0.08)**	**8.65 (0.23)**	**13.34 (0.13)**
FIXED	2.76 (0.14)	9.54 (0.15)	14.03 (0.18)
SCOOT	2.43 (0.16)	9.94 (0.52)	19.24 (0.38)
SUPRL	2.78 (0.11)	8.96 (0.26)	14.76 (0.13)

TABLE 8.6: Average traffic density (ATD) for each mechanism and traffic scenario.

Table 8.6 shows that the performance of DC2 in terms of ATD is similar to some of the other benchmarks. In *unstructured* traffic, the majority of mechanism have low ATD, below 5 *vpk*. Additionally, the difference in ATD between DC2 and the benchmarks is small in *unstructured* traffic. Figure 8.18a shows that the ATD of DC2 is only significantly different to FIXED and SUPRL in *unstructured* traffic. In the *football* scenario, there is a small but significant difference between the ATD of the market-based mechanisms and the benchmarks. DC2 has lower ATD than FIXED, SCOOT and SUPRL in the *football* scenario. This is also true in the *directional* traffic scenario. In *structured* traffic, DC2 has lower ATD than SCOOT and SUPRL but not FIXED. In *regional* traffic, DC2 outperforms FIXED but not SCOOT and SUPRL. Lastly, in *constant* traffic, DC2 has lower ATD than SCOOT and SUPRL but not FIXED. In all three scenarios with *predictable* traffic, the difference between DC2 and the benchmarks is significant, see Figure 8.18.



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

(e) Directional

(f) Constant

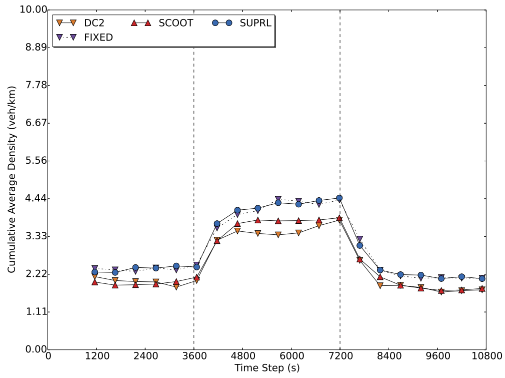
Figure 8.18: Visual representation of two-sample Mann-Whitney test conducted on ATD (Portland map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

8.3.5 Cumulative Average Density (CAD)

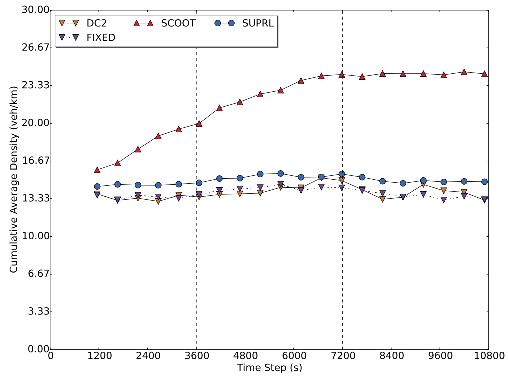
In the *unstructured* scenario, all of the mechanisms have low CAD. Figure 8.19a shows that the CAD of DC2 is similar to SCOOT and SUPRL. Both mechanisms have lower CAD than FIXED and SUPRL. In the *football* scenario, differences amongst the mechanisms do not really appear until during the football match and the second disruption, see Figure 8.19c. During the first disruptions all the mechanisms have a similar rate of increase in CAD, however, the mechanisms display different recovery periods during the match. For example, the CAD of DC2, FIXED and SUPRL declines faster than SCOOT during the football match. During the second disruption, the mechanisms display another rise in CAD but the CAD of DC2 and SUPRL remains lower than FIXED and SCOOT.

In *directional*, *structured* and *constant* traffic, DC2 outperforms SCOOT and SUPRL, see Figure 8.19. In both traffic scenarios, DC2 shows little change in CAD during the disruptions. However, Figure 8.19b, shows that in *structured* traffic, DC2 performs similar to FIXED. Additionally, in *constant* traffic, the difference between SUPRL and DC2 is negligible at times. DC2 also has low CAD in *directional* and *structured* traffic on the Phoenix map.

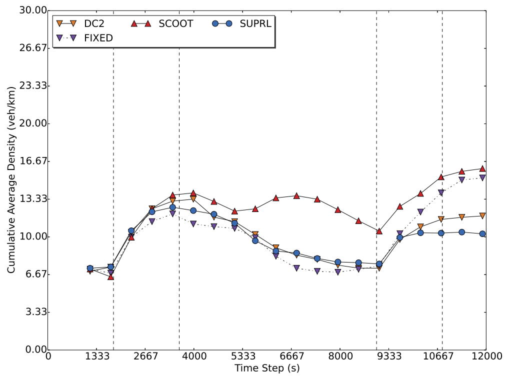
Finally, in *regional* traffic, DC2 have lower CAD than FIXED. Prior to the *regional* traffic disruption, DC2 and SUPRL have similar CAD, however, during the disruption the CAD of DC2 increases and remains higher than the CAD of SUPRL and SCOOT for the remainder of the scenario.



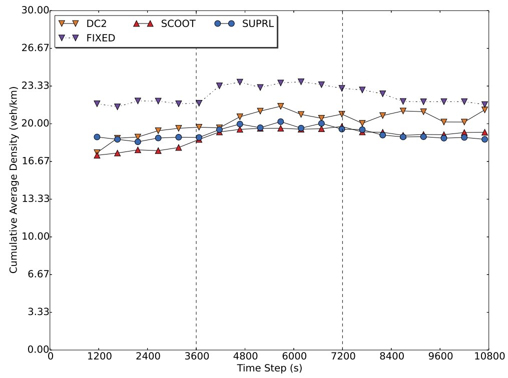
(a) Unstructured.



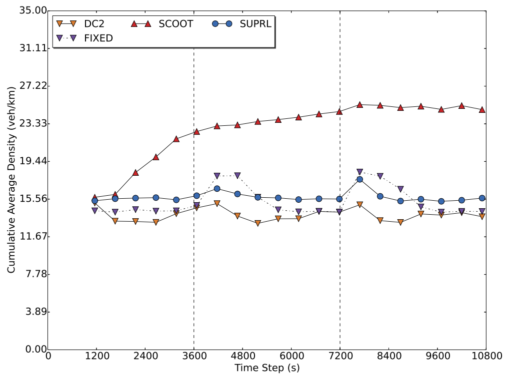
(b) Structured.



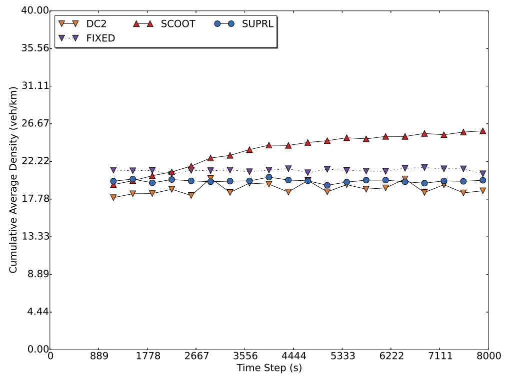
(c) Football



(d) Regional



(e) Directional

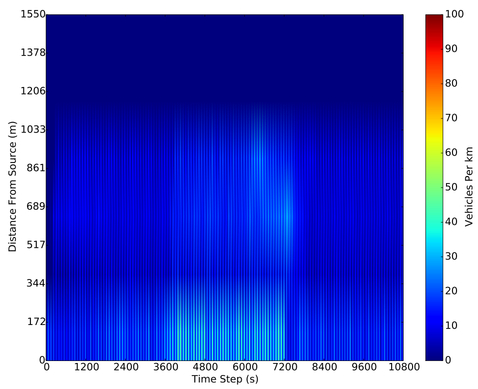


(f) Constant

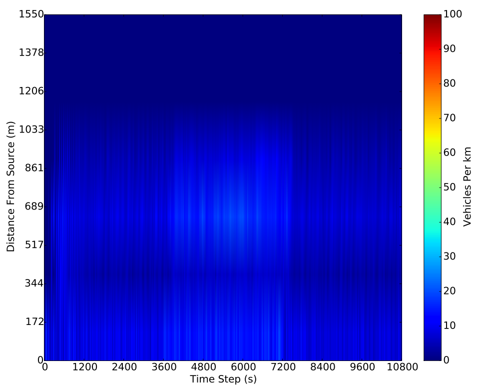
FIGURE 8.19: Cumulative average density (over 30 simulations) on the Portland map. Beginning and ending of disruptions are marked by dotted lines.

8.3.6 Density on Major Artery

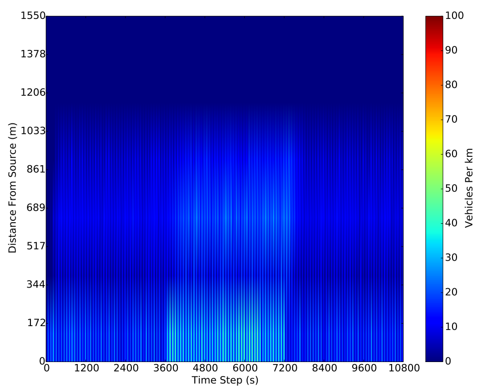
In *unstructured* traffic, Figure 8.20 shows that there is little difference amongst the mechanisms in terms of traffic density on the second artery. All of the mechanisms show low amounts of traffic on the second artery. Figure 8.20 also shows that DC2 has slightly more traffic, on the first road segment of the second artery, than SCOOT. In the *football* scenario, the traffic density on the second artery for DC2 is less intense and occurs for a shorter period of time in comparison to SCOOT and FIXED, see Figure 8.21. Also, in comparison to SUPRL, DC2 has higher traffic density on the first and second road segments of the second artery. DC2, SCOOT, and SUPRL have similarities in traffic density on the fourth artery in the *directional* traffic scenario, see Figure 8.22. That is, all three mechanisms have high traffic density on the first road segment which dissipates downstream. Also, in *directional* traffic, DC2 has much higher traffic density on the first road segment than SCOOT and initially, DC2 has several downstream road segments with high levels of traffic density.



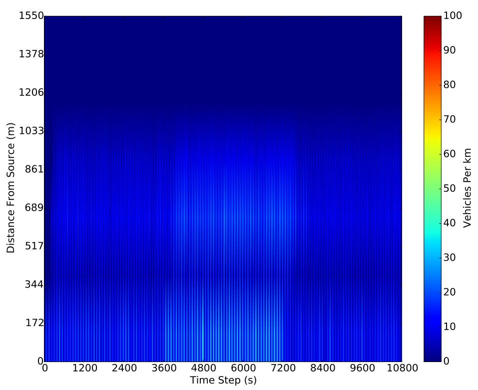
(a) DC2



(b) SCOOT

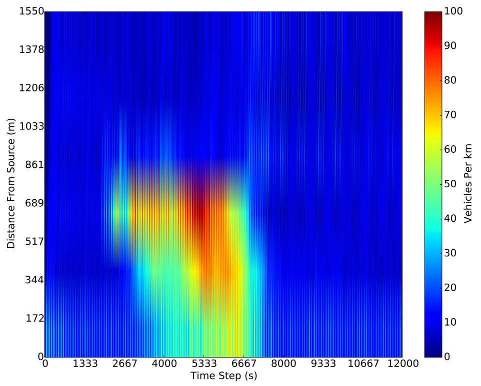


(c) SUPRL

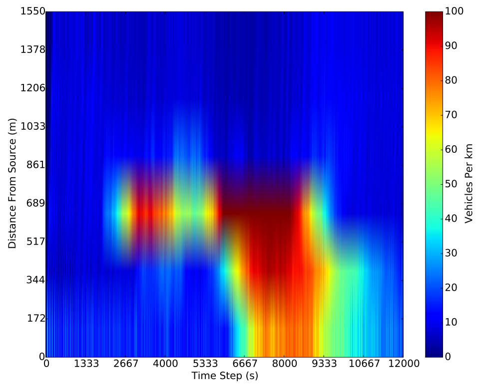


(d) FIXED

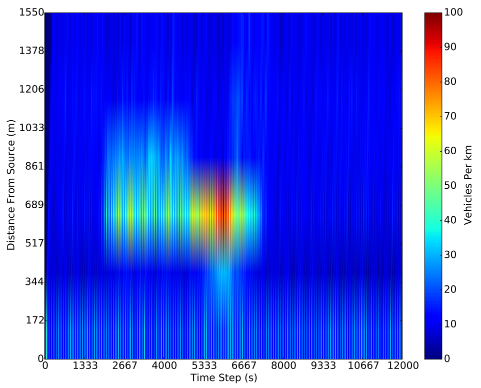
FIGURE 8.20: Traffic density on major artery in *unstructured* traffic on the Portland map.



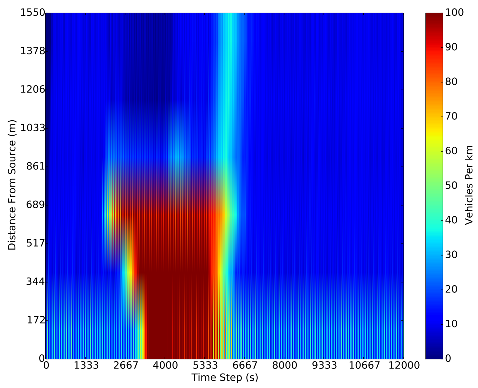
(a) DC2



(b) SCOOT

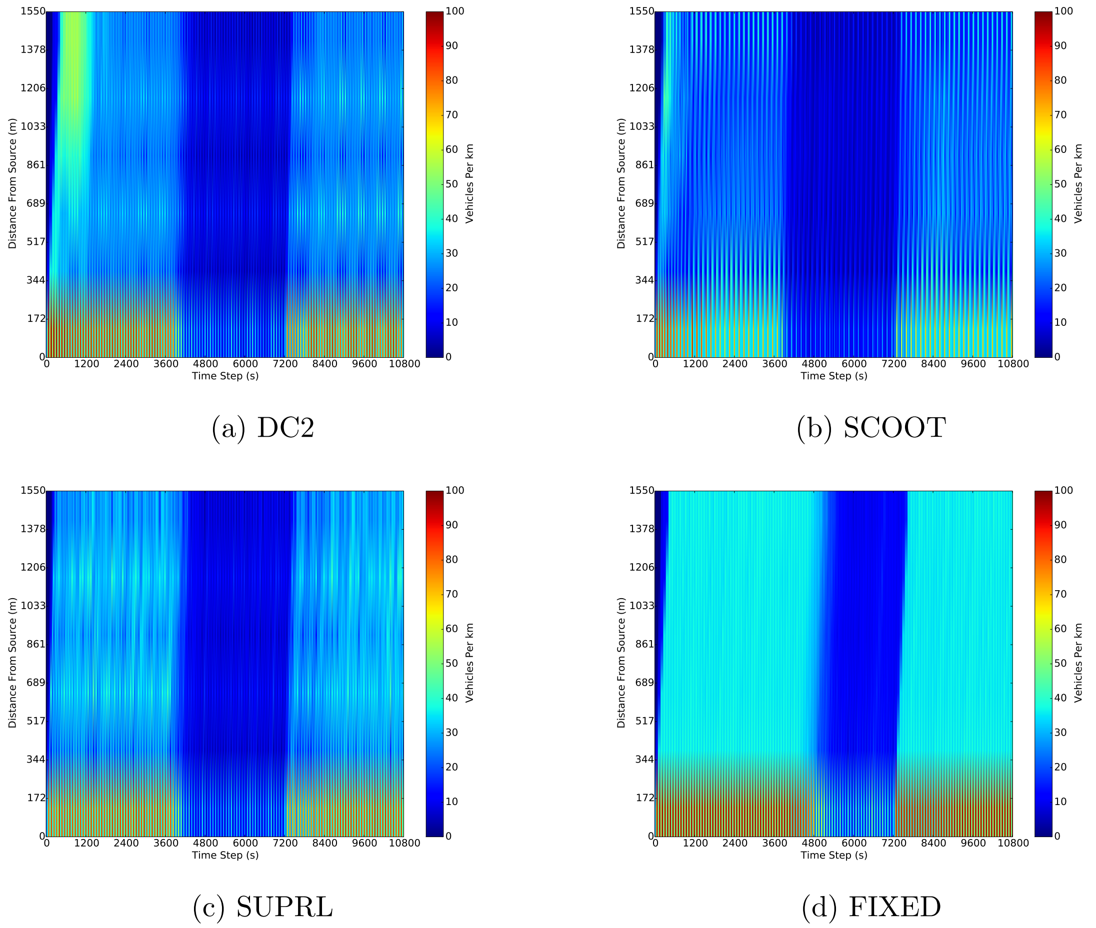


(c) SUPRL



(d) FIXED

FIGURE 8.21: Traffic density on major artery in *football* traffic on the Portland map.



(a) DC2

(b) SCOOT

(c) SUPRL

(d) FIXED

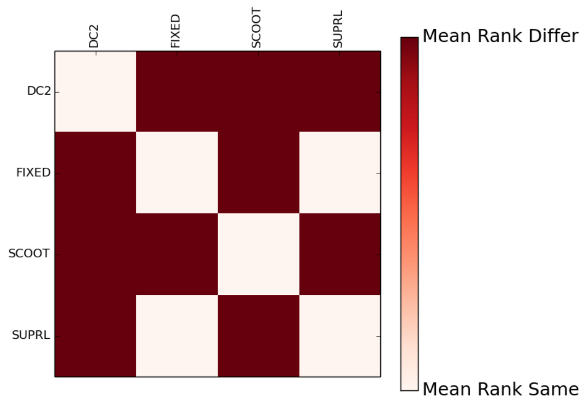
FIGURE 8.22: Traffic density on major artery in *directional* traffic on Portland map.

Average Number of Stops (ANS) (*std.*)			
		Traffic Pattern	
Mechanism	*Structured*	*Regional*	*Constant*
DC2	170 (2.49)	234.83 (7.93)	**232.93 (4.79)**
FIXED	**161.13 (2.71)**	280.5 (2.71)	279.97 (2.54)
SCOOT	319.8 (13.78)	203.77 (7.74)	347.6 (19.31)
SUPRL	185.33 (3.18)	**201.1 (2.68)**	246 (3.45)
	Unstructured	*Football*	*Directional*
DC2	22.13 (1.59)	131.07 (5.68)	**163 (2.18)**
FIXED	28.83 (2.45)	143.4 (3.61)	187.1 (2.5)
SCOOT	**19.27 (2.27)**	153.77 (13.37)	344.93 (7.87)
SUPRL	29.63 (1.52)	**122.73 (4.89)**	194.37 (2.34)

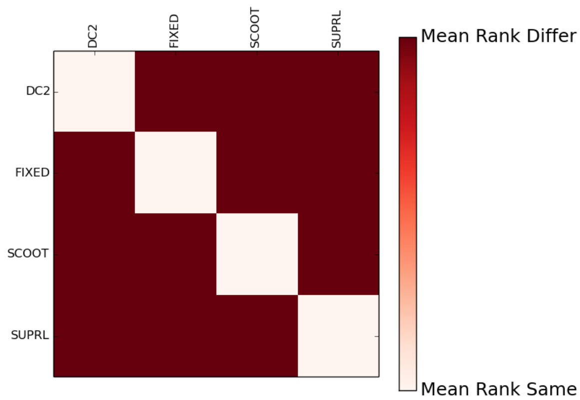
Table 8.7: Average *number of stops* (ANS) for each mechanism and traffic scenario.

8.3.7 Vehicle Stops (ANS)

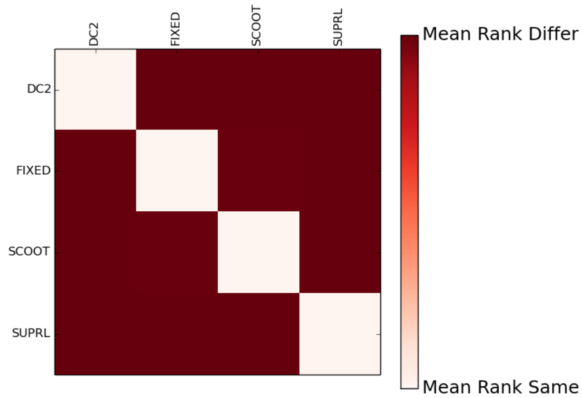
Overall, the *unstructured* traffic scenario has fewer stops in comparison to the other scenarios, Table 8.7. Table 8.7 also shows that DC2 has lower ANS than FIXED and SUPRL in the *unstructured* traffic scenario. However, in *unstructured* traffic on the Portland map, SCOOT has the lowest ANS, this is not the case on the Phoenix map. In the *football* scenario, DC2 has lower ANS than FIXED and SCOOT but not SUPRL. The difference in ANS performance between DC2 and the benchmarks in scenarios with *unpredictable* traffic flow is significant, see Figure 8.23. In *structured* traffic, DC2 has lower ANS to SCOOT and SUPRL only. Also, in *regional* traffic, DC2 has lower ANS than FIXED but not SCOOT and SUPRL, whereas in *constant* traffic DC2 has lower ANS than all three benchmarks. The difference between the ANS of DC2 and the benchmarks is significant, Figure 8.23, in *structured* and *regional* traffic.



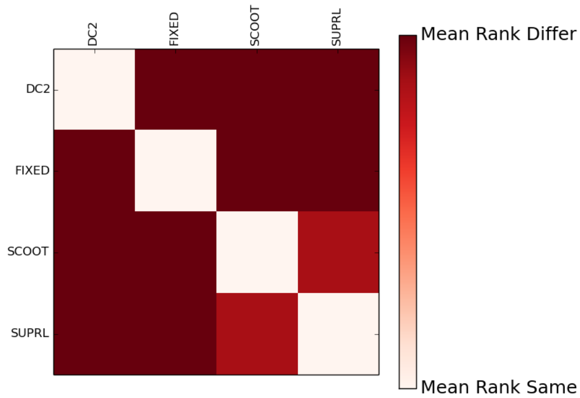
(a) Unstructured.



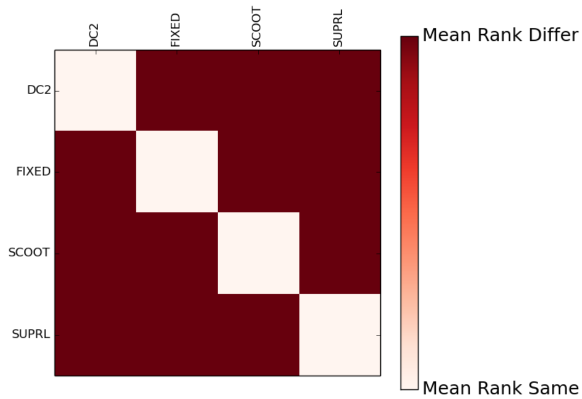
(b) Structured.



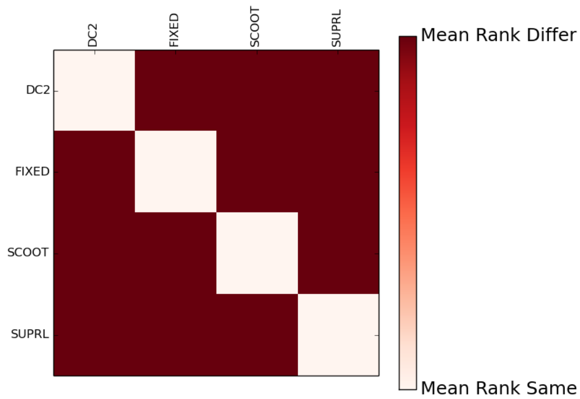
(c) Football



(d) Regional



(e) Directional



(f) Constant

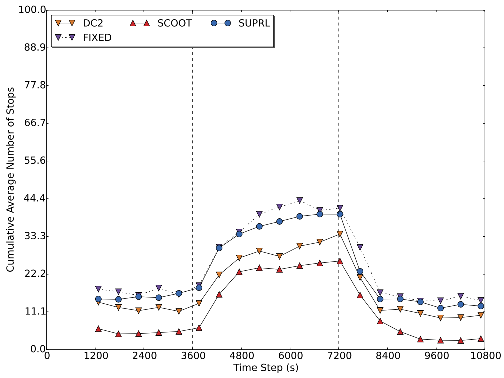
FIGURE 8.23: Visual representation of two-sample Mann-Whitney test conducted on ANS (Portland map) results from the 30 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

8.3.8 Cumulative Average Number of Stops (CANS)

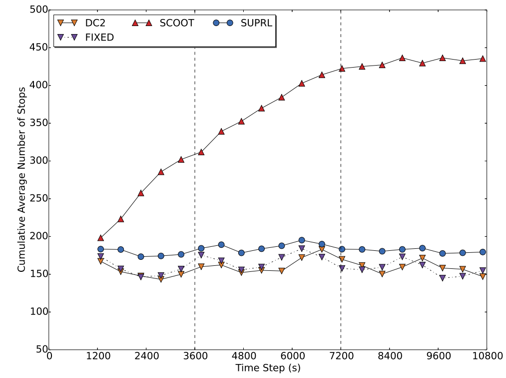
In *unstructured* traffic, DC2 have lower CANS than SUPRL and FIXED, even during the disruption, see Figure 8.24a. However, in *unstructured* traffic, SCOOT has the lowest CANS during the entire scenario. On the Portland map the *unstructured* traffic disruption does not affect the mechanisms in the same manner as it does on the Phoenix map. In *unstructured* traffic on the Portland map, during the disruption, the increase in CANS displayed by DC2, FIXED, SCOOT, and SUPRL is not as sharp as it on the Phoenix map.

In the first disruption of the *football* scenario, all the mechanisms have similar rates of increase in CANS. DC2, FIXED and SUPRL have a much sharper decline in CANS during the match than SCOOT. Additionally, during the same period, DC2 performs similar to SUPRL in terms of CANS. Although at the beginning of the football match the CANS of DC2 is similar to SCOOT's CANS, by the end of the football the CANS of DC2 is much lower than the CANS of SCOOT. Also, during the second disruption DC2 and SUPRL are able to maintain lower numbers of stopped vehicles compared to SCOOT and FIXED.

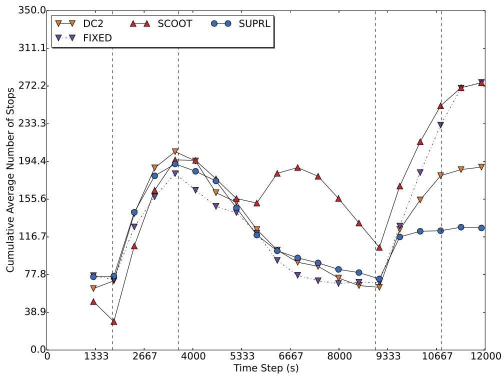
In *structured* and *directional* traffic, DC2 outperforms SCOOT and SUPRL in maintaining low CANS, see Figure 8.24. In both scenarios, DC2 does not display a substantial increase in CANS during the disruptions. However, in *structured* traffic, the CANS of DC2 is similar to FIXED. In *regional* traffic, DC2 have higher CANS than SCOOT and SUPRL but lower CANS than FIXED. In the *constant* traffic scenario, Figure 8.24f, the CANS of DC2 is lower than FIXED and SCOOT.



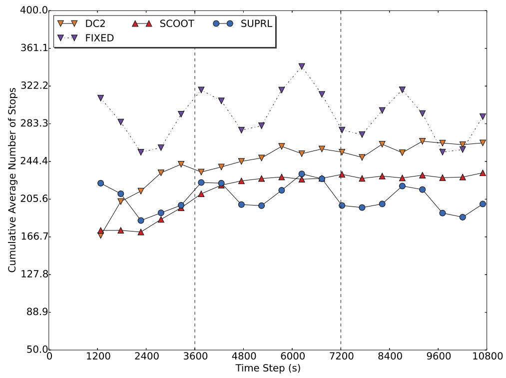
(a) Unstructured.



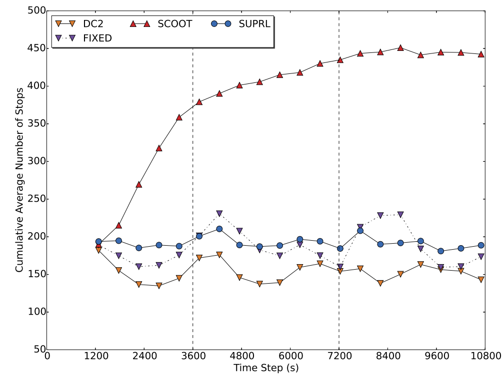
(b) Structured.



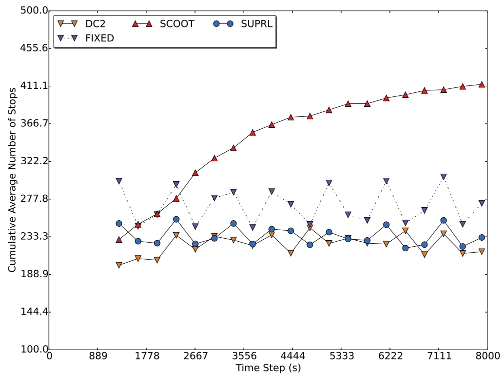
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 8.24: Cumulative average number of stops (over 30 simulations) on the Portland map. Beginning and ending of disruptions are marked by dotted lines.

8.4 Summary

This section presents a summary of my findings for DC2. The ATT, ATD and ANS results discussed in this section are aggregated across both maps.

8.4.1 ATT

	Average Travel Time (ATT) (*std.*)		
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
DC2	**187.21 (40.08)**	201.12 (51.64)	**214.96 (26.24)**
FIXED	197.75 (31.92)	237.7 (54.1)	225.27 (41.02)
SCOOT	252.83 (109.24)	**183.36 (54.52)**	215.81 (72)
SUPRL	201.8 (42.7)	195.23 (51.65)	229.62 (24.35)
	Unstructured	*Football*	*Directional*
DC2	**499.7 (19.74)**	227.93 (71.56)	**185.26 (36.45)**
FIXED	839.34 (296.72)	245.25 (55.66)	208.08 (35.21)
SCOOT	870.98 (446.74)	288.77 (106.27)	258.03 (112.15)
SUPRL	716.68 (150.72)	**207.16 (65.13)**	204.78 (44.77)

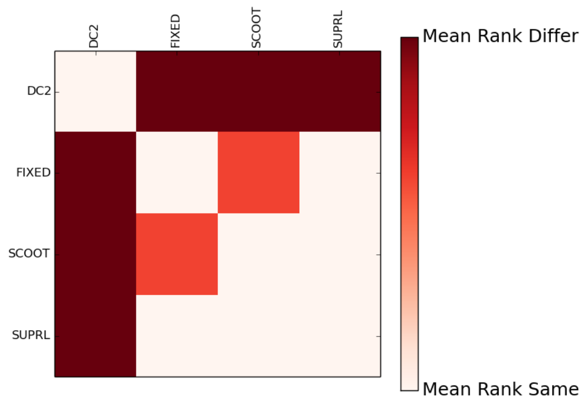
Table 8.8: Average travel times (ATT) for each mechanism and traffic scenario.

Table 8.8 contains the ATT across both maps for the mechanisms evaluated in this chapter. DC2 outperforms FIXED, SCOOT and SUPRL in *unstructured*, *directional*, *structured*, and *constant* traffic. However, in *football* and *regional* traffic, DC2 is outperformed by SUPRL and SCOOT, respectively.

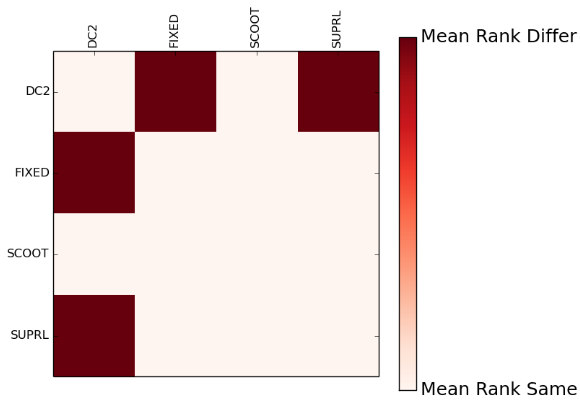
The ATT results show that allowing intersections to influence their neighbours preferences will result in a significant difference in the ATT of DC2 compared with SCOOT, SUPRL and FIXED based on traffic conditions.

Hypothesis 16 *There will be a significant difference in the* ATT *of DC2 compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

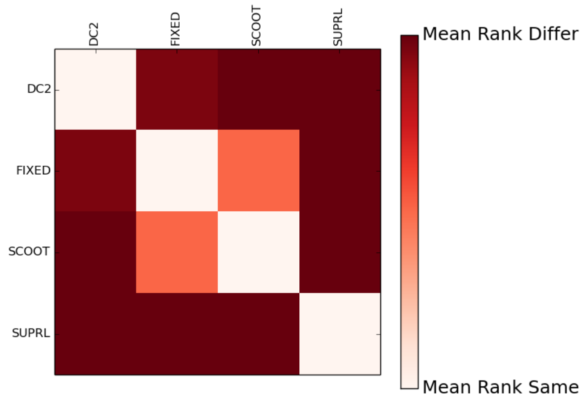
—In *unstructured*, *football* and *regional* traffic the difference between FIXED, SCOOT and SUPRL is significant; in *directional* and *structured* traffic, the difference in ATT is only significant to the ATT of FIXED and SUPRL. Lastly, in *constant* traffic, the ATT of DC2 is only significantly different to the ATT of SUPRL.



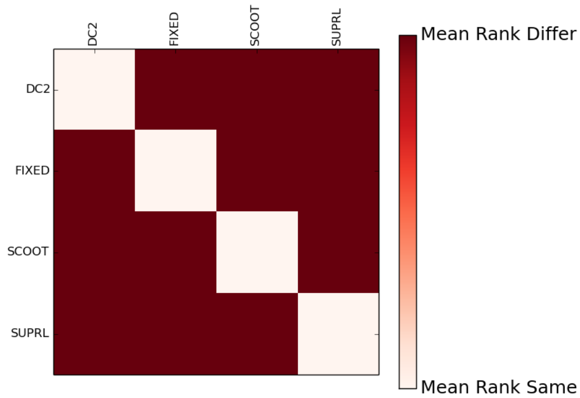
(a) Unstructured.



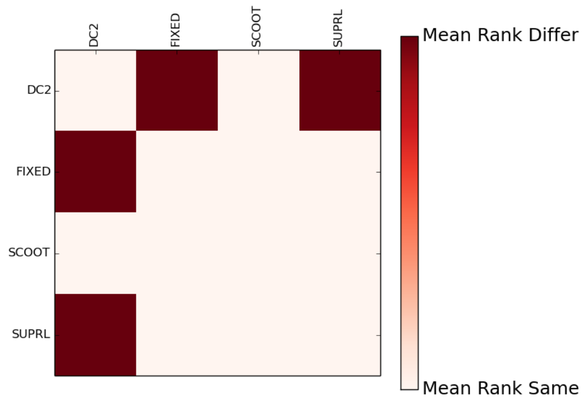
(b) Structured.



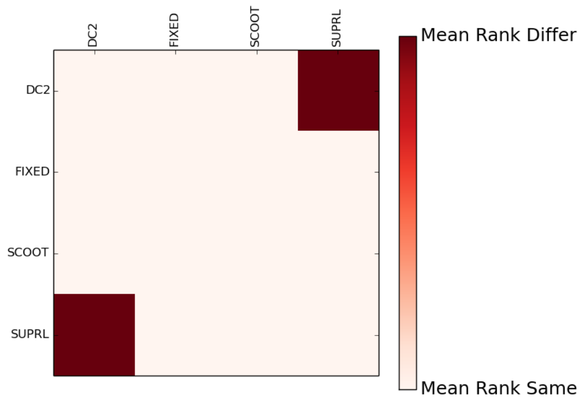
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 8.25: Visual representation of two-sample Mann-Whitney test conducted on ATT results from the 60 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

8.4.2 ATD

Average Traffic Density (ATD) (*std.*)			
	Traffic Pattern		
Mechanism	*Structured*	*Regional*	*Constant*
DC2	**12.33 (1.12)**	15.84 (2.63)	21.08 (2.57)
FIXED	12.41 (0.2)	18.34 (2.15)	21.06 (0.4)
SCOOT	14.66 (3.64)	**14.35 (2.68)**	**20.37 (2.39)**
SUPRL	13.05 (0.92)	15.24 (2.17)	22.52 (3.11)
	Unstructured	*Football*	*Directional*
DC2	**3.79 (1.46)**	7.7 (0.98)	**12.64 (0.71)**
FIXED	6.61 (3.99)	8.81 (0.82)	13.76 (0.31)
SCOOT	7.11 (5.3)	8.73 (1.3)	15.49 (3.79)
SUPRL	5.66 (2.96)	**7.55 (1.45)**	13.8 (0.98)

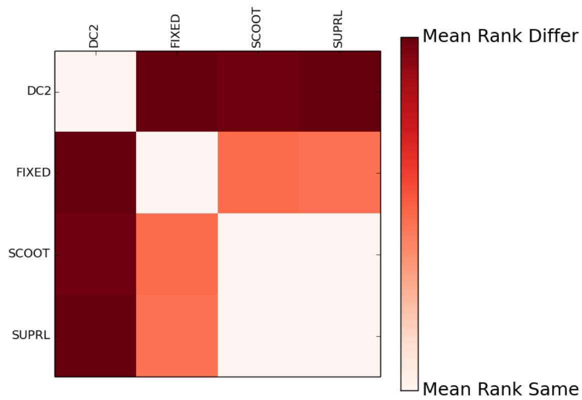
TABLE 8.9: Average traffic density (ATD) for each mechanism and traffic scenario.

Table 2 contains the ATT across both maps for the mechanisms evaluated in this chapter. DC2 outperforms FIXED, SCOOT and SUPRL in *structured*, *unstructured* and *directional* traffic. Similar to the ATT results, SCOOT and SUPRL outperform DC2 in *regional* and *football*, respectively. Lastly, in *constant* traffic SCOOT outperforms DC2.

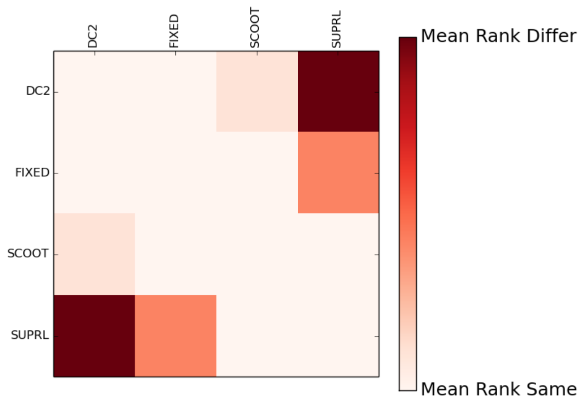
The ATD results show that allowing intersections to influence their neighbours preferences will result in a significant difference in the ATD of DC2 compared with SCOOT, SUPRL and FIXED based on traffic conditions.

Hypothesis 17 *There will be a significant difference in* ATD *of DC2 compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

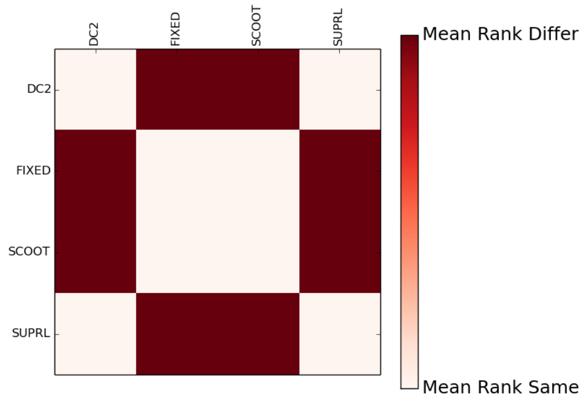
—In *unstructured* and *regional* traffic, the ATD of DC2 is significantly different from FIXED, SCOOT and SUPRL; in *football* traffic, the ATD of DC2 is significantly different from FIXED and SCOOT; in *structured* traffic, the ATD of DC2 is significantly different from SUPRL; in *directional* traffic, the ATD of DC2 is significantly different from FIXED and SUPRL; and in *constant* traffic, the ATD of DC2 is significantly different from SCOOT and SUPRL.



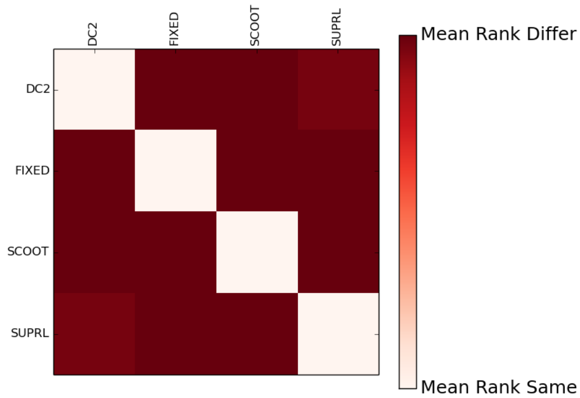
(a) Unstructured.



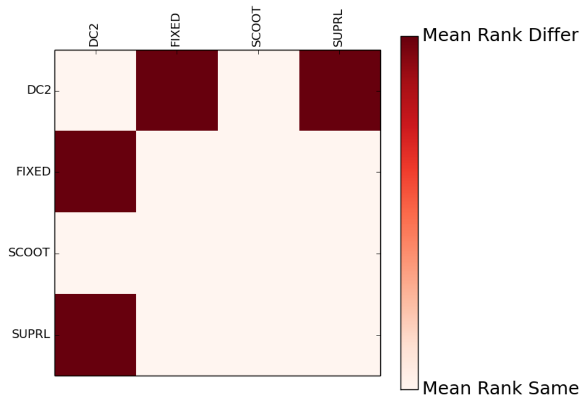
(b) Structured.



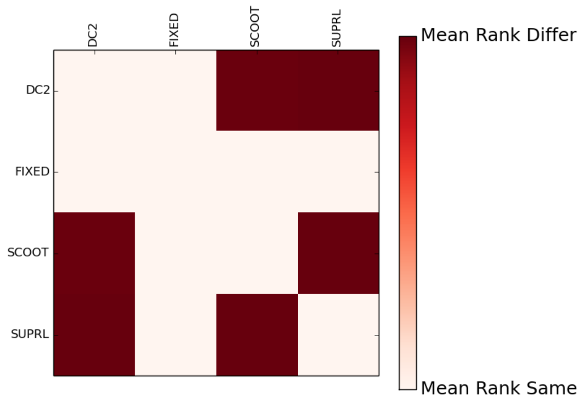
(c) Football



(d) Regional



(e) Directional



(f) Constant

FIGURE 8.26: Visual representation of two-sample Mann-Whitney test conducted on ATD results from the 60 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

Average Number of Stops (ANS) (*std.*)			
		Traffic Pattern	
Mechanism	*Structured*	*Regional*	*Constant*
DC2	**114.13 (56.37)**	154.4 (81.43)	**192.27 (41.45)**
FIXED	115.68 (45.88)	189.45 (91.85)	205.33 (75.29)
SCOOT	189.98 (131.28)	**128.87 (75.75)**	223.12 (126.29)
SUPRL	125.98 (59.9)	132.07 (69.65)	203.12 (43.92)
	Unstructured	*Football*	*Directional*
DC2	**20.83 (2.02)**	84.98 (46.68)	**112.82 (50.63)**
FIXED	48.58 (23.13)	96.83 (47.16)	133.03 (54.56)
SCOOT	54.62 (51.38)	96.13 (58.91)	204.75 (141.49)
SUPRL	38.12 (10.22)	**76.43 (46.83)**	131.62 (63.31)

TABLE 8.10: Average *number of stops* (ANS) for each mechanism and traffic scenario.

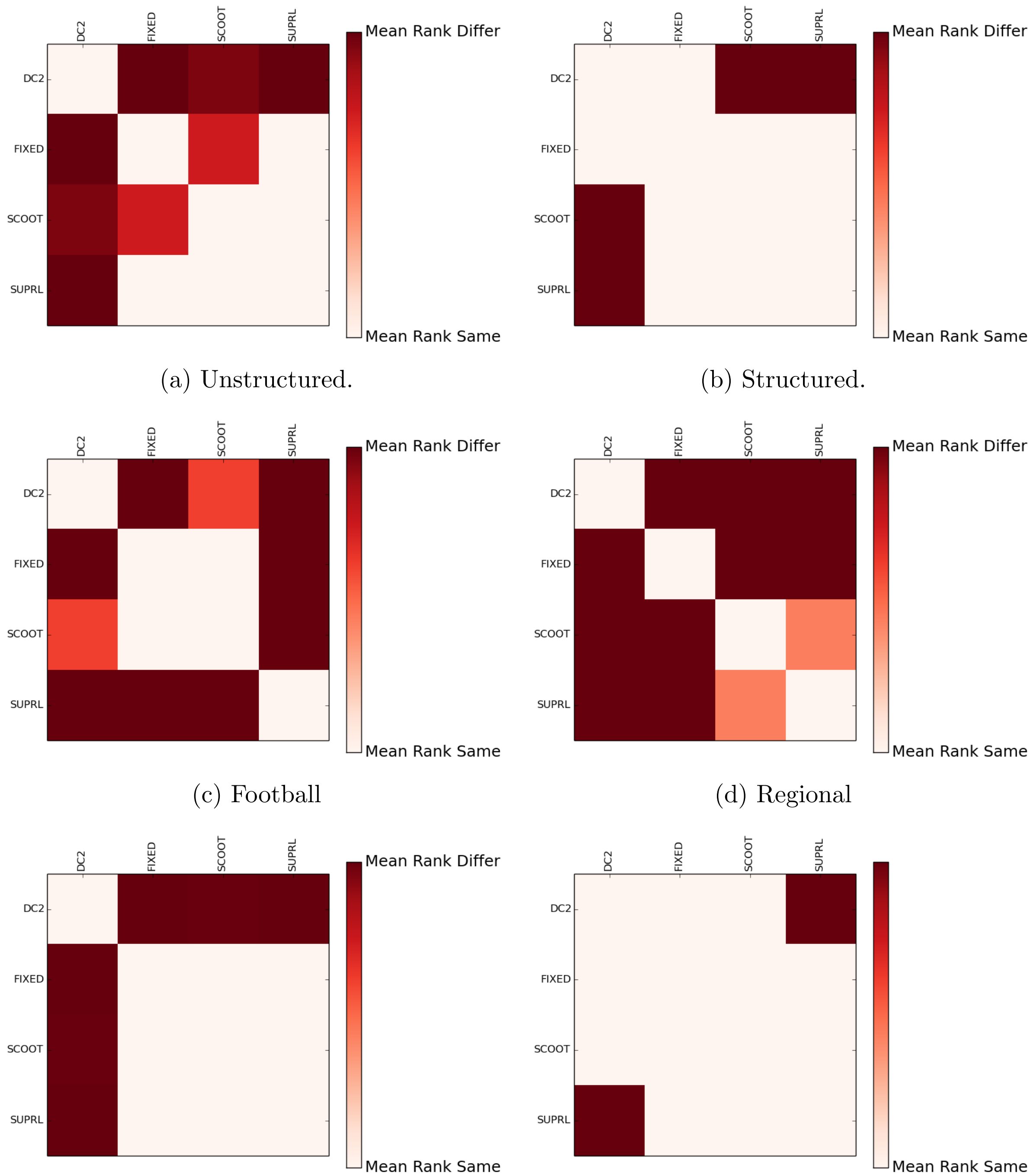
8.4.3 ANS

Table 8.10 contains the ATT across both maps for the mechanisms evaluated in this chapter. The ANS results are similar to the ATT results. DC2 outperforms FIXED, SCOOT and SUPRL in *unstructured*, *directional*, *structured* and *constant* traffic. However, in *regional* and *football* traffic, SCOOT and SUPRL outperform DC2, respectively.

The ANS results show that allowing intersections to influence their neighbours preferences will result in a significant difference in the ANS of DC2 compared with SCOOT, SUPRL and FIXED based on traffic conditions.

Hypothesis 18 *There will be a significant difference in* ANS *of DC2 compared with SCOOT, SUPRL and FIXED based on traffic conditions.*

—In *unstructured*, *directional* and *regional* traffic, the ANS of DC2 is significantly different from FIXED, SCOOT and SUPRL; in *football* traffic, the ANS of DC2 is significantly different from FIXED and SUPRL; in *structured* traffic, the ANS of DC2 is significantly different from SCOOT and SUPRL; and in *constant* traffic, the ATD of DC2 is significantly different from SUPRL only.



(a) Unstructured.

(b) Structured.

(c) Football

(d) Regional

(e) Directional

(f) Constant

Figure 8.27: Visual representation of two-sample Mann-Whitney test conducted on ANS results from the 60 simulations for each mechanism in each scenario. The p-value from each test is represented as a coloured square, where dark squares denote statistical significance.

The findings shows that allowing intersections to influence their neighbours preferences will result in a significant difference in the performance of DC2 compared with SCOOT, SUPRL and FIXED depending on the traffic scenario. Furthermore, in terms of ATT of ANS, DC2 outperforms the benchmarks in the majority of traffic scenarios. In *unstructured structured*, *directional* and *constant* traffic, DC2 has lower ATT and ANS than all three benchmarks.

Chapter 9

Conclusion

9.1 Discussion

This section discusses the summary of results from Chapters 6, 7, and 8. The results are discussed in the following order: evaluation of GRACE in Section 9.1.1, evaluation of the traffic control parameters in Section 9.1.2 and finally, DC2 in Section 9.1.3.

9.1.1 GRACE Mechanisms

SAT and SATQ have several characteristics in common; both mechanisms adjust the *split* in five second segments. In other words, the amount of *green* time allotted to a phase either increases or decreases by five seconds, depending on which of the two traffic signal agents won the auction. Furthermore, SAT/Q does not adjust *cycle* or *offset*. Despite their similarities, the results for the two mechanisms are quite different.

Results show that the performance of SATQ depends on the traffic conditions while SAT performs poorly in all six traffic scenarios. SATQ has lower ATT than all three benchmarks (FIXED, SCOOT and SUPRL) in *unstructured* traffic and the difference in performance is significant. Also, in *unstructured* traffic, SATQ has lower ANS than SUPRL (the difference in ANS of SATQ and SUPRL is significant). Although the difference in performance between SATQ and SCOOT is not always significant, SATQ does outperform SCOOT in terms of ATT and ANS in traffic scenarios other than *unstructured* traffic, e.g., *football* and *directional* traffic. SAT, on the other hand, has higher ATT, ATD and ANS than SATQ and the benchmarks, with few exceptions. For example, in *regional*, SAT has lower ATD than SATQ.

The main difference between SAT and SATQ is that the utility function of traffic signal agents in SATQ have an added term which represents the *fullness* of the road segment the signal agent serves. That is, this term is akin to a vehicle count of the road segment. The performance of SATQ suggests that this added term provides a better picture of traffic conditions and enables SATQ to adjust the allocation of green time accordingly. SUPRL works with similar traffic information. However, in SUPRL agents utilise vehicle queue size, i.e., the number of vehicles that have come to a complete stop.

Additionally, SATQ's performance in *unstructured* traffic also suggests that, in some traffic conditions, unconnected intersections (i.e., SATQ does not coordinate signal timing with neighbours) may outperform systems that attempt to coordinate intersections.

The performance of the GRACE variant, MMDOS, depends on the traffic condition as well. Unlike SAT/Q which only adjust green time, MMDOS adjusts all three traffic control parameters, *split*, *cycle* and *offset*. Moreover, MMDOS does not utilise vehicle counts or road *fullness* as SATQ does. Instead, MMDOS relies on vehicle detectors to provide data on traffic volume and estimation of vehicle stops. MMDOS outperforms SAT in all six traffic scenarios regardless of the performance metric, but the difference between SAT and MMDOS is not always significant. MMDOS outperforms SATQ in five of the traffic scenarios (SATQ has lower ATT, ATD, and ANS than MMDOS in *unstructured* traffic), but again the difference between SATQ and MMDOS is not always significant. The difference in performance between SAT/Q and MMDOS is most significant in terms of ANS where MMDOS has lower ANS than SAT/Q in *football*, *directional*, *constant*, and *regional* traffic. The MMDOS results suggest that adjusting more than just *split* can improve the performance of my market-based traffic controller, however, this depends on the traffic conditions.

Although SUPRL is designed to reduce stops, SUPRL does not have the lowest ANS in all of the traffic scenarios. In the *constant* traffic scenario, MMDOS has the lowest ANS and the difference between the ANS of MMDOS and the ANS of SUPRL is significant. Although both MMDOS and SUPRL attempt to reduce stops, they do so in different ways. SUPRL relies solely on *green* time to reduce the number of stops. That is, SUPRL can increase the *green* time of an intersection; this, in turn, gives certain vehicles approaching the intersection more time to reach the intersection before a red phase begins. Simply put, the number of stops can be reduced by increasing the green phase of a series of intersections (assuming all green phases are feeding traffic in a single direction). However, this has the added side-effect of creating an imbalance in the allocation of *green* time at the intersection(s). MMDOS and other mechanisms that manipulate the *offset* of traffic signals are able to reduce stops without having this side effect. That is, in MMDOS, the difference in the starting time of the cycle of two adjacent intersections is adjusted in order to give vehicles, travelling from one intersection to the next, time to reach the second intersection while the intersection is in the right phase (the *right* phase meaning the phase which allows the vehicle to traverse the intersection without stopping). MMDOS results suggest that this alternative option (as MMDOS is also able to adjust *green* time) for reducing stops is advantageous in some traffic conditions, e.g., *constant* traffic.

9.1.2 Traffic Control Parameters

This thesis also evaluates the impact of three key traffic control parameters: *split*, *cycle* and *offset*. Variants of MMDOS and SCOOT, which utilised different combinations of *split*, *cycle* and *offset*, were implemented to evaluate the traffic control parameters. The evaluation of the traffic control parameters demonstrates the effects that the three traffic control parameters have on ATT, ATD and ANS. More specifically, the results show that adopting alternative combinations of *split*, *cycle* and *offset* may improve traffic performance depending on traffic conditions. Additionally, differences in performance based on the adjustment of different sets of traffic control parameters are present in SCOOT as well as MMDOS.

The traffic control parameter which produces the lowest ATT, ATD and ANS in one traffic scenario does not have the same results across all the other traffic scenarios. The variance in performance suggests that the control parameter (i.e., the instrument used to manage traffic flow, e.g., *split* or *cycle*) is as important as the traffic control mechanism itself. Regardless of the internal structure and processes of the mechanism, if it uses traffic signals, then the traffic signal becomes an end point. That is, the traffic signal is an interface between the mechanism and the traffic flow that must be managed. Although the traffic signal operates on simple rules, it defines the final solution space of the traffic control mechanism. Therefore, traffic control parameter(s) should be selected with care and calculation.

In every traffic scenario, there is a variant of MMDOS and SCOOT which outperforms the original version, i.e., MMDOS and SCOOT —this does however depend on the traffic conditions. The evaluation of the traffic control parameters suggest that some combinations of parameters fair better in *predictable* traffic than in *unpredictable* traffic. Mechanisms that adjust *cycle*-only (or *cycle* and *offset*) tend to perform better in *predictable* traffic, e.g., *structured*, *regional* and *constant* traffic. The greater the cycle length, the greater the capacity of the intersection. That is, greater capacity increases the intersection's ability to handle large volumes of traffic without reaching congestion levels. In the *cycle*-only mechanisms, increases in *cycle* length are achieved by adding more time to the *green* portions of the *cycle*. This increases effective *green* time while the total lost time (the amber and red portions of the cycle) remains the same. Thus, with *predictable* traffic flows, the network demands are easily met with increased cycle lengths. Long cycle lengths can also be a source of delay. Longer cycles impose longer wait times for those vehicles that are waiting for their turn (during their red phase) to use the intersection. In the scenarios with *predictable* traffic flows, cross traffic is low. Hence, there is less conflict of movements at intersections, which mitigates the negative impact of increasing the *cycle* length. In *unpredictable* traffic flows, however, traffic demand cannot be met with *cycle* changes alone. In *unpredictable* traffic flows, the allocation of *green* time is far more important as demonstrated by the performance of MMDOS(SO) and SCOOT(SO) in *unstructured* traffic. *Directional* traffic may seem as an exception, the MMDOS variant with the lowest ATT in *directional* traffic is MMDOS(C). However, in *directional* traffic, there are periods during the scenario where the traffic flow can be characterised as *predictable*. In the *directional* traffic scenario, there is less change in the cross traffic than in *unstructured* traffic. The same is true during the football match in the *football* traffic scenario. MMDOS(C) performance is most likely due to the fact that both *football* and *directional* traffic have periods where traffic conditions are ideal for *cycle* only adjustments.

The evaluation of the traffic control parameters also shows that the same effects (variability in performance) found in the MMDOS strategies are also present in the SCOOT variants. More importantly, in every traffic scenario, there is a SCOOT variant that outperforms SCOOT. The improvements in ATT of SCOOT(C) over SCOOT in the *structured* traffic suggest that relaxing some of the restrictions on the control parameters that SCOOT uses may improve its performance. SCOOT is designed to optimise the signal timing of small sets of traffic signals that form a linear path. This severely restricts the ability of SCOOT to adapt to unexpected cross traffic. SCOOT performs well with traffic that has some established pattern of behaviour but can not cope in *unpredictable* traffic, e.g., the *unstructured* and *football* scenarios. SCOOT has the lowest ATT in *regional* and *constant* traffic, although, in *constant* traffic the difference between SCOOT and the other mechanisms is not significant.

Certain combinations of traffic control parameters appear to perform in a similar manner. In every traffic scenario, x(S) and x(SO) have similar ATT, ATD and ANS. The same is true for x(SC) and MMDOS (which adjusts *split*, *cycle* and *offset*) as well as x(C) and x(CO). This suggests that *split* and *cycle* have a greater influence on traffic performance than *offset*. The *offset* parameter is more difficult to adjust than the other traffic control parameters because it relies on certain properties of traffic, mainly the formation of platoons. *Offsets* are sensitive to average travel times, specifically between intersections; even small changes in platoon speeds can cause *offset* adjustments to increase delays instead of reducing travel times.

9.1.3 DC2

DC2 adjusted only the *split* and *offset*. However, unlike MMDOS and SAT/Q, traffic signal agents in DC2 are influenced by neighbouring intersections. More specifically, traffic signal agents in DC2 prefer *offset* adjustments that improve the overall performance of the intersection, not just the vehicle movement(s) the traffic signal agent represents. Also, DC2 relies on vehicle detectors to provide data on traffic volume and estimation of vehicle stops just as MMDOS does. DC2 has lower ATT, ATD and ANS than all three benchmarks in *unstructured* traffic. DC2 also has lower ATT, ATD and ANS than all three benchmarks in *directional* traffic, however, the difference in the ATT and ATD of DC2 and the benchmarks is not significant. Despite the performance of DC2 in *unstructured* and *directional* traffic, the results are not far from the ATT and ANS results for MMDOS variants that adjust *split* only. The ATT and ANS performance of DC2 supports the conclusion that *offset* adjustments are the least effective of the three traffic control parameters for improving traffic flow.

9.2 Conclusion

This thesis described an approach to market-based traffic control systems which deviates from traditional approaches where vehicles (and the transportation infrastructure) require additional technological capabilities. More specifically, the proposed approach is free of vehicle agents (or other embedded software) which are often used to participate in auctions on behalf of the driver and in some applications control the vehicle. Removing vehicle agents from the control loop lessens communication demands on the transportation infrastructure, that is, without vehicle agents $V2I$ or $V2V$ communications are no longer an issue. The market-based traffic control system proposed in this thesis is far more practical than those described in the market-based traffic control literature. This thesis has adds to the discourse on the applications of markets within the the traffic domain.

9.2.1 Contributions

This thesis makes five contributions:

I. This thesis demonstrated the efficacy of my multi-agent auction-based traffic control framework which allows for the use of an auction mechanism within the traffic control system without vehicle agents. The market-based traffic control mechanisms adjusted different combinations of *split*, *cycle* and *offset*. SAT and SATQ adjusted *green* time only and the GRACE (MMDOS and its variants) adjusted different combinations of the three traffic control parameters. The performance of MMDOS and SATQ, especially with *unstructured* traffic, demonstrates the ability of my market-based traffic control systems to lower travel times and stops.

II. This thesis demonstrated the efficacy of dynamic coalition formation within my market-based traffic control system. In DC2, preferences of traffic signal agents (i.e., preferred signal timing adjustments) are influenced by neighbouring intersections. In DC2, dynamic coalitions are formed between pairs of intersections during which time the *split* is adjusted as well as the *offset* in order to reduce vehicle stops. Although only slightly better than the MMDOS variants, DC2 displays an improvement in all three performance metrics. Again the difference in performance is most notable in *unstructured* traffic, where DC2 has lower ATT, ATD and ANS than all three benchmarks.

III. Managing traffic flow using traffic signals is by far the most prevalent method of traffic control. More importantly, this means that traffic at signalised intersections is managed with a handful of parameters, e.g., phase order or a combination of the parameters studied in this thesis. Consequently, our understanding of traffic signals is strictly defined by the traffic control parameter(s) selected by the engineer within a specific context (e.g., neural network based traffic controller). Therefore, the relationship between traffic signal timing and traffic performance is not entirely clear. The importance of evaluating all the combinations of *split*, *cycle* and *offset* is that this analysis presents an alternative possibility for improving an existing traffic control system. More specifically, the analysis conducted in this thesis has important implications for SCOOT. This thesis provides evidence that SCOOT's performance may be increased by simply adjusting subset combinations of *split*, *cycle* and *offset*. Furthermore, similarities between the principles used for signal timing adjustment in my market-based traffic controller and SCOOT suggest that loosening some SCOOT restrictions may improve performance as well, specifically, static *regions*.

IV. This thesis provides an in-depth description of SCOOT, a popular commercial adaptive traffic control system. The chapter which covers SCOOT includes common default values for SCOOT parameters and and algorithms describing the procedures that take place when *split*, *cycle* and *offset* are adjusted in SCOOT. More importantly, this thesis includes a number of sources to gain further understanding on the principles behind SCOOT and how it has evolved over the past three decades.

V. Experiments were conducted in six traffic scenarios, on two maps, using three performance metrics (*average travel time*, *average traffic density* and *average number of stops*). The traffic scenarios represented a variety of road traffic conditions. Additionally, five of the traffic scenarios have disruptions (an increase in traffic intensity or a change in the direction of the heaviest traffic flow). All of the traffic control systems, SAT/Q, the GRACE variants, SCOOT and SUPRL showed some variability in performance, i.e, how well the traffic controller performed, depended on the traffic scenario. The specific details of traffic flow scenarios are shared in the Appendix, as a contribution to provide standard experiment environments for the research community.

9.2.2 Future Work

The evaluation of the traffic control systems in this thesis was conducted in a traffic simulator, which is a common practice for testing traffic control systems. This presents a number of limitations. This thesis primarily focuses on the impact of the intersection control policy on traffic performance. Thus, experiments are conducted with homogeneous control policies, i.e., signalised intersections without any other type of intersection control policies such as yields, stop signs and roundabouts. Additionally, traffic simulations do not include pedestrians, cyclists, public transport, emergency response vehicles, or any other modes of transportation other than passenger vehicles.

Future work will address limitations of the simulation environment and explore new methods for improving performance:

I. Although there are many cities around the world that have grid-based road networks, replicating experiments on a larger road network, specifically one that has evolved more organically than a grid-based network, would better demonstrate the efficacy of my approach. As well as using road networks that are not grid-based, it would also be beneficial to model a city which utilises SCOOT. The aim is to have a road network(s) modelled in SUMO which includes SCOOT region(s), a mixture of intersection control policies (e.g., non-signalised intersections and roundabouts) and vehicles other than cars (e.g., public transport and lorries). Lastly, this new simulated traffic environment should include signalised intersections that possess more than two phases. Additional phases in the signal plan will help to shed light on how well my approach works on traffic signals with more complex traffic sequences.

II. The multi-agent framework used in my approach is amenable to machine learning techniques. First, one component of the traffic control system, in particular, has been identified as an area which may function better with the use of machine learning; that component is the bidding strategy. Second, an analysis of my results has revealed a possible opportunity to further improve the system with machine learning.

   (a) Traffic signal agents use a *fixed* bidding strategy, more specifically, it is strictly a function of the traffic conditions on the incoming links of the intersection. Machine learning may be used to discover temporal aspects of bidding which are not currently exploited in the present bidding strategy. That is, it is possible that the bidding strategy should change depending on not just traffic demands but also the time of day.

   (b) Results show that the performance of different combinations of traffic control parameters depends on the traffic conditions. For example, adjusting *cycle* only (or *cycle* and *offset*) offers better performance in *predictable* traffic. Hence, a learning technique can be used to learn which combinations of traffic control parameters the auction should adjust given prevailing traffic conditions.

III. In some problem domains that have employed auctions for resource (or task) allocation, it is often clear when an auction should occur. For example, in multi-robot systems, where auctions are used for task allocation, whenever a new task is up for completion the system should execute an auction to determine which robot will complete the task. However, it is not always the case that employing auctions will also reveal *when* the auction should occur. In my auction-based traffic control system, auctions are executed periodically (e.g., every five minutes) to facilitate adjustments to traffic signal timing. Moreover, the manner in which the auction is employed in my approach does not provide a natural means of determining when the intersection agents should execute an auction. This presents an interesting area for further research; the study of alternative methods for initiating auctions in in my traffic control system as well as other applications of auctions where it may not be clear when auctions should occur.

IV. The DC2 variant forms dynamic coalitions which allow intersections to better coordinate traffic signal timing through *offset* adjustments. However, the coalitions are of size two, that is, the coalition facilitates coordination between only two intersections. Future work aims to develop a method for creating coalitions that involve more than two intersections. One of the obstacles in forming larger coalitions is defining a beneficial relationship between intersections that are separated by more than one link. Furthermore, this new process for coalition formation must work within the constraints of the traffic domain as well as the auction framework. Hence, a possible solution is to include a pricing mechanism for traffic signal adjustments. In this way, intersections are able to influence the signal timing adjustments of other distant intersections (not just their immediate neighbours).

V. Lastly, future work will also investigate the effects of network topology on the performance of my traffic control system. As mentioned previously, all the road networks used in my experiments were grid-based. The evaluation of my approach on a variety of road network types would provide further evidence of its capabilities as an alternative means of tackling traffic control.

9.3 Summary

This thesis focused on the use of market mechanisms for traffic control, more specifically, the application of market principles set forth in market-based multi-robot systems to the traffic domain. Thus, the primary goal of this thesis was to design, implement and evaluate a novel multi-agent market-based traffic control system which does not rely on *vehicle agents* and other major changes to vehicles or transportation infrastructure. Evaluation of the traffic control system was conducted on two grid-based maps using six different traffic scenarios. The traffic scenarios represented various traffic conditions which included changes in traffic intensity and direction. The traffic scenarios were simulated in SUMO, an open source, macro traffic simulator. Additionally, performance was measured using three metrics: *travel time*, *traffic density*, and *number of stops*.

This thesis made five contributions: (*i*) demonstrated the efficacy of a novel multi-agent market-based traffic control methodology; (*ii*) demonstrated the efficacy of a market-based technique for dynamic coalition formation; (*iii*) analysed three key traffic control parameters used by SCOOT; (*iv*) developed of a Python implementation of SCOOT for use on SUMO and (*v*) conducted a thorough evaluation of the novel market-based mechanisms, along with SCOOT and a reinforcement-learning traffic controller, over a variety of road traffic conditions. This thesis provided a unique insight into the behaviour of three key traffic control parameters and results showed that the novel market-based mechanism has the potential to improve traffic performance in traffic conditions that are less than ideal for SCOOT.

Appendix A

Traffic Routes: Phoenix

Section 5.2 includes a brief explanation of how traffic is generated in SUMO. In SUMO, vehicles traverse pre-set paths and the volume of traffic (measured in *passenger car unit* per hour or *pcu/hour*) are specified for each path. This appendix shows the paths and traffic volume used in each traffic scenario on the Phoenix map.

***structured* Traffic on Phoenix Map**	
Demand (*pcu/hour*)	Route
597	(1,0),(1,1),(1,2),(1,3),(1,4)
227	(1,4),(1,3),(1,2),(1,1),(1,0)
576	(2,0),(2,1),(2,2),(2,3),(2,4)
245	(2,4),(2,3),(2,2),(2,1),(2,0)
223	(3,0),(3,1),(3,2),(3,3),(3,4)
590	(3,4),(3,3),(3,2),(3,1),(3,0)
220	(0,1),(1,1),(2,1),(3,1),(4,1)
234	(4,1),(3,1),(2,1),(1,1),(0,1)
241	(0,2),(1,2),(2,2),(3,2),(4,2)
248	(4,2),(3,2),(2,2),(1,2),(0,2)
238	(0,3),(1,3),(2,3),(3,3),(4,3)
230	(4,3),(3,3),(2,3),(1,3),(0,3)

TABLE A1: Routes and traffic demand for the *structured* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0, 0)$.

***unstructured* Traffic on Phoenix Map**	
Demand (*pcu/hour*)	Route
130	(4,0),(4,1),(3,1),(3,2),(2,2),(2,3),(3,3),(4,3),(4,4),(3,4), (2,4),(1,4),(1,3),(0,3),(0,2),(1,2), (1,1),(1,0),(0,0)
126	(0,0),(0,1),(0,2),(1,2),(1,1),(2,1),(3,1),(3,0),(4,0),(4,1), (4,2),(4,3),(4,4),(3,4),(3,3),(2,3),(1,3),(1,4),(0,4)
115	(0,4),(1,4),(1,3),(0,3),(0,2),(0,1),(0,0),(1,0),(2,0),(2,1), (2,2),(2,3),(2,4),(3,4),(3,3),(3,2),(4,2),(4,3),(4,4)
140	(0,4),(1,4),(2,4),(3,4),(4,4),(4,3),(4,2),(4,1),(3,1),(3,2), (3,3),(2,3),(2,2),(1,2),(0,2),(0,1),(1,1),(1,0),(2,0),(3,0),(4,0)

Table A2: Routes and traffic demand for the *unstructured* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0, 0)$.

***football* Traffic on Phoenix Map**	
Demand (*pcu/hour*)	Route
227	(1,0),(1,1),(1,2),(1,3),(1,4)
220	(1,4),(1,3),(1,2),(1,1),(1,0)
216	(2,0),(2,1),(2,2),(2,3),(2,4)
223	(2,4),(2,3),(2,2),(2,1),(2,0)
245	(3,0),(3,1),(3,2),(3,3),(3,4)
234	(3,4),(3,3),(3,2),(3,1),(3,0)
90	(0,1),(1,1),(2,1),(3,1),(4,1)
97	(4,1),(3,1),(2,1),(1,1),(0,1)
79	(0,2),(1,2),(2,2),(3,2),(4,2)
83	(4,2),(3,2),(2,2),(1,2),(0,2)
104	(0,3),(1,3),(2,3),(3,3),(4,3)
86	(4,3),(3,3),(2,3),(1,3),(0,3)
292†	(0,2),(1,2),(2,2)
288†	(4,2),(3,2),(2,2)
306†	(0,4),(0,3),(1,3),(2,3),(2,2)
299†	(4,0),(4,1),(3,1),(3,2),(2,2)
295††	(2,2),(1,2),(0,2)
320††	(2,2),(3,2),(4,2)
313††	(2,2),(2,3),(1,3),(0,3),(0,4)
310††	(2,2),(3,2),(3,1),(4,1),(4,0)

Table A3: Routes and traffic demand for the *football* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0, 0)$. The football scenario has two disruptions. In the first disruption traffic enters the city (†) and the second traffic exits the city (††).

***directional* Traffic on Phoenix Map**	
Demand (*pcu/hour*)	Route
705	(1,0),(1,1),(1,2),(1,3),(1,4)
226	(1,4),(1,3),(1,2),(1,1),(1,0)
684	(2,0),(2,1),(2,2),(2,3),(2,4)
244	(2,4),(2,3),(2,2),(2,1),(2,0)
223	(3,0),(3,1),(3,2),(3,3),(3,4)
698	(3,4),(3,3),(3,2),(3,1),(3,0)
219	(0,1),(1,1),(2,1),(3,1),(4,1)
234	(4,1),(3,1),(2,1),(1,1),(0,1)
241	(0,2),(1,2),(2,2),(3,2),(4,2)
248	(4,2),(3,2),(2,2),(1,2),(0,2)
237	(0,3),(1,3),(2,3),(3,3),(4,3)
230	(4,3),(3,3),(2,3),(1,3),(0,3)

TABLE A4: Routes and traffic demand for the *directional* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0, 0)$.

***regional* Traffic on Phoenix Map**	
Demand (*pcu/hour*)	Route
597	(1,0),(1,1),(1,2),(1,3),(1,4)
586	(1,4),(1,3),(1,2),(1,1),(1,0)
576	(2,0),(2,1),(2,2),(2,3),(2,4)
604	(2,4),(2,3),(2,2),(2,1),(2,0)
590	(3,0),(3,1),(3,2),(3,3),(3,4)
594	(3,4),(3,3),(3,2),(3,1),(3,0)
147	(0,1),(1,1),(2,1),(3,1),(4,1)
162	(4,1),(3,1),(2,1),(1,1),(0,1)
169	(0,2),(1,2),(2,2),(3,2),(4,2)
176	(4,2),(3,2),(2,2),(1,2),(0,2)
165	(0,3),(1,3),(2,3),(3,3),(4,3)
158	(4,3),(3,3),(2,3),(1,3),(0,3)

TABLE A5: Routes and traffic demand for the *regional* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0, 0)$.

constant **Traffic on Phoenix Map**	
Demand (*pcu/hour*)	Route
540	(1,0),(1,1),(1,2),(1,3),(1,4)
540	(1,4),(1,3),(1,2),(1,1),(1,0)
540	(2,0),(2,1),(2,2),(2,3),(2,4)
540	(2,4),(2,3),(2,2),(2,1),(2,0)
540	(3,0),(3,1),(3,2),(3,3),(3,4)
540	(3,4),(3,3),(3,2),(3,1),(3,0)
216	(0,1),(1,1),(2,1),(3,1),(4,1)
216	(4,1),(3,1),(2,1),(1,1),(0,1)
216	(0,2),(1,2),(2,2),(3,2),(4,2)
216	(4,2),(3,2),(2,2),(1,2),(0,2)
216	(0,3),(1,3),(2,3),(3,3),(4,3)
216	(4,3),(3,3),(2,3),(1,3),(0,3)

TABLE A6: Routes and traffic demand for the *constant* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0, 0)$.

Appendix B

Traffic Routes: Portland

Section 5.2 includes a brief explanation of how traffic is generated in SUMO. In SUMO, vehicles traverse pre-set paths and the volume of traffic (measured in *passenger car unit* per hour or *pcu/hour*) are specified for each path. This appendix shows the paths and traffic volume used in each traffic scenario on the Portland map.

unstructured **Traffic on Portland Map**	
Demand (*pcu/hour*)	Route
129	(4,0),(4,1),(3,1),(3,2),(2,2),(2,3),(3,3),(4,3),(4,4),(3,4),(2,4),(1,4),(1,3),(1,2),(1,1),(1,0)
126	(2,0),(2,1),(1,1),(1,2),(2,2),(3,2),(4,2),(4,3),(4,4),(5,4),(5,5),(5,6),(4,6),(3,6),(2,6),(1,6),(0,6)
115	(0,5),(1,5),(1,4),(1,3),(1,2),(1,1),(2,1),(2,2),(2,3),(2,4),(3,4),(3,3),(3,2),(4,2),(4,3),(4,4),(4,5),(5,5),(6,5),(6,6),(6,7)
140	(0,4),(1,4),(2,4),(3,4),(4,4),(4,3),(4,2),(4,1),(3,1),(3,2),(3,3),(2,3),(2,2),(1,2),(1,1),(2,1),(3,1),(4,1),(5,1),(6,1),(7,1)

Table A1: Routes and traffic demand for the *unstructured* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0, 0)$.

***structured* Traffic on Portland Map**	
Demand (*pcu/hour*)	Route
223	(1,0),(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(1,7)
234	(2,0),(2,1),(2,2),(2,3),(2,4),(2,5),(2,6),(2,7)
230	(3,0),(3,1),(3,2),(3,3),(3,4),(3,5),(3,6),(3,7)
223	(4,0),(4,1),(4,2),(4,3),(4,4),(4,5),(4,6),(4,7)
241	(5,0),(5,1),(5,2),(5,3),(5,4),(5,5),(5,6),(5,7)
223	(6,0),(6,1),(6,2),(6,3),(6,4),(6,5),(6,6),(6,7)
219	(1,7),(1,6),(1,5),(1,4),(1,3),(1,2),(1,1),(1,0)
223	(2,7),(2,6),(2,5),(2,4),(2,3),(2,2),(2,1),(2,0)
234	(3,7),(3,6),(3,5),(3,4),(3,3),(3,2),(3,1),(3,0)
248	(4,7),(4,6),(4,5),(4,4),(4,3),(4,2),(4,1),(4,0)
237	(5,7),(5,6),(5,5),(5,4),(5,3),(5,2),(5,1),(5,0)
237	(6,7),(6,6),(6,5),(6,4),(6,3),(6,2),(6,1),(6,0)
241	(7,1),(6,1),(5,1),(4,1),(3,1),(2,1),(1,1),(0,1)
241	(7,2),(6,2),(5,2),(4,2),(3,2),(2,2),(1,2),(0,2)
248	(7,3),(6,3),(5,3),(4,3),(3,3),(2,3),(1,3),(0,3)
234	(7,4),(6,4),(5,4),(4,4),(3,4),(2,4),(1,4),(0,4)
237	(7,5),(6,5),(5,5),(4,5),(3,5),(2,5),(1,5),(0,5)
219	(7,6),(6,6),(5,6),(4,6),(3,6),(2,6),(1,6),(0,6)
230	(0,1),(1,1),(2,1),(3,1),(4,1),(5,1),(6,1),(7,1)
241	(0,2),(1,2),(2,2),(3,2),(4,2),(5,2),(6,2),(7,2)
601	(0,3),(1,3),(2,3),(3,3),(4,3),(5,3),(6,3),(7,3)
601	(0,4),(1,4),(2,4),(3,4),(4,4),(5,4),(6,4),(7,4)
576	(0,5),(1,5),(2,5),(3,5),(4,5),(5,5),(6,5),(7,5)
601	(0,6),(1,6),(2,6),(3,6),(4,6),(5,6),(6,6),(7,6)

TABLE A2: Routes and traffic demand for the *structured* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0, 0)$.

football **Traffic on Portland Map**	
Demand (*pcu/hour*)	Route
223	(1,0),(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(1,7)
226	(2,0),(2,1),(2,2),(2,3),(2,4),(2,5),(2,6),(2,7)
226	(3,0),(3,1),(3,2),(3,3),(3,4),(3,5),(3,6),(3,7)
234	(4,0),(4,1),(4,2),(4,3),(4,4),(4,5),(4,6),(4,7)
241	(5,0),(5,1),(5,2),(5,3),(5,4),(5,5),(5,6),(5,7)
216	(6,0),(6,1),(6,2),(6,3),(6,4),(6,5),(6,6),(6,7)
230	(1,7),(1,6),(1,5),(1,4),(1,3),(1,2),(1,1),(1,0)
216	(2,7),(2,6),(2,5),(2,4),(2,3),(2,2),(2,1),(2,0)
226	(3,7),(3,6),(3,5),(3,4),(3,3),(3,2),(3,1),(3,0)
219	(4,7),(4,6),(4,5),(4,4),(4,3),(4,2),(4,1),(4,0)
219	(5,7),(5,6),(5,5),(5,4),(5,3),(5,2),(5,1),(5,0)
248	(6,7),(6,6),(6,5),(6,4),(6,3),(6,2),(6,1),(6,0)
82	(7,1),(6,1),(5,1),(4,1),(3,1),(2,1),(1,1),(0,1)
104	(7,2),(6,2),(5,2),(4,2),(3,2),(2,2),(1,2),(0,2)
97	(7,3),(6,3),(5,3),(4,3),(3,3),(2,3),(1,3),(0,3)
97	(7,4),(6,4),(5,4),(4,4),(3,4),(2,4),(1,4),(0,4)
86	(7,5),(6,5),(5,5),(4,5),(3,5),(2,5),(1,5),(0,5)
72	(7,6),(6,6),(5,6),(4,6),(3,6),(2,6),(1,6),(0,6)
72	(0,1),(1,1),(2,1),(3,1),(4,1),(5,1),(6,1),(7,1)
100	(0,2),(1,2),(2,2),(3,2),(4,2),(5,2),(6,2),(7,2)
82	(0,3),(1,3),(2,3),(3,3),(4,3),(5,3),(6,3),(7,3)
93	(0,4),(1,4),(2,4),(3,4),(4,4),(5,4),(6,4),(7,4)
100	(0,5),(1,5),(2,5),(3,5),(4,5),(5,5),(6,5),(7,5)
72	(0,6),(1,6),(2,6),(3,6),(4,6),(5,6),(6,6),(7,6)
$295^{\dagger}$	(0,2),(1,2),(2,2),(2,3),(3,3),(3,4)
$313^{\dagger}$	(7,1),(6,1),(6,2),(5,2),(4,2),(3,2),(2,2),(2,3),(3,3),(3,4)
$298^{\dagger}$	(0,5),(1,5),(1,4),(2,4),(3,4)
$295^{\dagger}$	(3,0),(3,1),(3,2),(2,2),(2,3),(2,4),(3,4)
$313^{\dagger\dagger}$	(3,4),(3,3),(2,3),(2,2),(1,2),(0,2)
$295^{\dagger\dagger}$	(3,4),(3,3),(2,3),(2,2),(3,2),(4,2),(5,2),(6,2),(6,1),(7,1)
$291^{\dagger\dagger}$	(3,4),(2,4),(1,4),(1,5),(0,5)
$320^{\dagger\dagger}$	(3,4),(2,4),(2,3),(2,2),(3,2),(3,1),(3,0)

TABLE A3: Routes and traffic demand for the *football* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0, 0)$. The football scenario has two disruptions. In the first disruption traffic enters the city (†) and the second traffic exits the city (††).

***directional* Traffic on Portland Map**	
Demand (*pcu/hour*)	Route
705	(1,0),(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(1,7)
684	(2,0),(2,1),(2,2),(2,3),(2,4),(2,5),(2,6),(2,7)
698	(3,0),(3,1),(3,2),(3,3),(3,4),(3,5),(3,6),(3,7)
698	(4,0),(4,1),(4,2),(4,3),(4,4),(4,5),(4,6),(4,7)
226	(5,0),(5,1),(5,2),(5,3),(5,4),(5,5),(5,6),(5,7)
244	(6,0),(6,1),(6,2),(6,3),(6,4),(6,5),(6,6),(6,7)
234	(1,7),(1,6),(1,5),(1,4),(1,3),(1,2),(1,1),(1,0)
241	(2,7),(2,6),(2,5),(2,4),(2,3),(2,2),(2,1),(2,0)
223	(3,7),(3,6),(3,5),(3,4),(3,3),(3,2),(3,1),(3,0)
219	(4,7),(4,6),(4,5),(4,4),(4,3),(4,2),(4,1),(4,0)
248	(5,7),(5,6),(5,5),(5,4),(5,3),(5,2),(5,1),(5,0)
237	(6,7),(6,6),(6,5),(6,4),(6,3),(6,2),(6,1),(6,0)
230	(0,1),(1,1),(2,1),(3,1),(4,1),(5,1),(6,1),(7,1)
226	(0,2),(1,2),(2,2),(3,2),(4,2),(5,2),(6,2),(7,2)
244	(0,3),(1,3),(2,3),(3,3),(4,3),(5,3),(6,3),(7,3)
223	(0,4),(1,4),(2,4),(3,4),(4,4),(5,4),(6,4),(7,4)
219	(0,5),(1,5),(2,5),(3,5),(4,5),(5,5),(6,5),(7,5)
234	(0,6),(1,6),(2,6),(3,6),(4,6),(5,6),(6,6),(7,6)
241	(7,1),(6,1),(5,1),(4,1),(3,1),(2,1),(1,1),(0,1)
248	(7,2),(6,2),(5,2),(4,2),(3,2),(2,2),(1,2),(0,2)
237	(7,3),(6,3),(5,3),(4,3),(3,3),(2,3),(1,3),(0,3)
230	(7,4),(6,4),(5,4),(4,4),(3,4),(2,4),(1,4),(0,4)
237	(7,5),(6,5),(5,5),(4,5),(3,5),(2,5),(1,5),(0,5)
230	(7,6),(6,6),(5,6),(4,6),(3,6),(2,6),(1,6),(0,6)

TABLE A4: Routes and traffic demand for the *directional* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0, 0)$.

***regional* Traffic on Portland Map**	
Demand (*pcu/hour*)	Route
583	(1,0),(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(1,7)
586	(2,0),(2,1),(2,2),(2,3),(2,4),(2,5),(2,6),(2,7)
594	(3,0),(3,1),(3,2),(3,3),(3,4),(3,5),(3,6),(3,7)
594	(4,0),(4,1),(4,2),(4,3),(4,4),(4,5),(4,6),(4,7)
608	(5,0),(5,1),(5,2),(5,3),(5,4),(5,5),(5,6),(5,7)
586	(6,0),(6,1),(6,2),(6,3),(6,4),(6,5),(6,6),(6,7)
604	(1,7),(1,6),(1,5),(1,4),(1,3),(1,2),(1,1),(1,0)
601	(2,7),(2,6),(2,5),(2,4),(2,3),(2,2),(2,1),(2,0)
604	(3,7),(3,6),(3,5),(3,4),(3,3),(3,2),(3,1),(3,0)
601	(4,7),(4,6),(4,5),(4,4),(4,3),(4,2),(4,1),(4,0)
583	(5,7),(5,6),(5,5),(5,4),(5,3),(5,2),(5,1),(5,0)
576	(6,7),(6,6),(6,5),(6,4),(6,3),(6,2),(6,1),(6,0)
165	(7,1),(6,1),(5,1),(4,1),(3,1),(2,1),(1,1),(0,1)
165	(7,2),(6,2),(5,2),(4,2),(3,2),(2,2),(1,2),(0,2)
151	(7,3),(6,3),(5,3),(4,3),(3,3),(2,3),(1,3),(0,3)
162	(7,4),(6,4),(5,4),(4,4),(3,4),(2,4),(1,4),(0,4)
172	(7,5),(6,5),(5,5),(4,5),(3,5),(2,5),(1,5),(0,5)
144	(7,6),(6,6),(5,6),(4,6),(3,6),(2,6),(1,6),(0,6)
144	(0,1),(1,1),(2,1),(3,1),(4,1),(5,1),(6,1),(7,1)
151	(0,2),(1,2),(2,2),(3,2),(4,2),(5,2),(6,2),(7,2)
144	(0,3),(1,3),(2,3),(3,3),(4,3),(5,3),(6,3),(7,3)
162	(0,4),(1,4),(2,4),(3,4),(4,4),(5,4),(6,4),(7,4)
176	(0,5),(1,5),(2,5),(3,5),(4,5),(5,5),(6,5),(7,5)
144	(0,6),(1,6),(2,6),(3,6),(4,6),(5,6),(6,6),(7,6)

TABLE A5: Routes and traffic demand for the *regional* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0,0)$.

***constant* Traffic on Portland Map**	
Demand (*pcu/hour*)	Route
540	(1,0),(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(1,7)
540	(2,0),(2,1),(2,2),(2,3),(2,4),(2,5),(2,6),(2,7)
540	(3,0),(3,1),(3,2),(3,3),(3,4),(3,5),(3,6),(3,7)
540	(4,0),(4,1),(4,2),(4,3),(4,4),(4,5),(4,6),(4,7)
540	(5,0),(5,1),(5,2),(5,3),(5,4),(5,5),(5,6),(5,7)
540	(6,0),(6,1),(6,2),(6,3),(6,4),(6,5),(6,6),(6,7)
540	(1,7),(1,6),(1,5),(1,4),(1,3),(1,2),(1,1),(1,0)
540	(2,7),(2,6),(2,5),(2,4),(2,3),(2,2),(2,1),(2,0)
540	(3,7),(3,6),(3,5),(3,4),(3,3),(3,2),(3,1),(3,0)
540	(4,7),(4,6),(4,5),(4,4),(4,3),(4,2),(4,1),(4,0)
540	(5,7),(5,6),(5,5),(5,4),(5,3),(5,2),(5,1),(5,0)
540	(6,7),(6,6),(6,5),(6,4),(6,3),(6,2),(6,1),(6,0)
216	(0,1),(1,1),(2,1),(3,1),(4,1),(5,1),(6,1),(7,1)
216	(0,2),(1,2),(2,2),(3,2),(4,2),(5,2),(6,2),(7,2)
216	(0,3),(1,3),(2,3),(3,3),(4,3),(5,3),(6,3),(7,3)
216	(0,4),(1,4),(2,4),(3,4),(4,4),(5,4),(6,4),(7,4)
216	(0,5),(1,5),(2,5),(3,5),(4,5),(5,5),(6,5),(7,5)
216	(0,6),(1,6),(2,6),(3,6),(4,6),(5,6),(6,6),(7,6)
216	(7,1),(6,1),(5,1),(4,1),(3,1),(2,1),(1,1),(0,1)
216	(7,2),(6,2),(5,2),(4,2),(3,2),(2,2),(1,2),(0,2)
216	(7,3),(6,3),(5,3),(4,3),(3,3),(2,3),(1,3),(0,3)
216	(7,4),(6,4),(5,4),(4,4),(3,4),(2,4),(1,4),(0,4)
216	(7,5),(6,5),(5,5),(4,5),(3,5),(2,5),(1,5),(0,5)
216	(7,6),(6,6),(5,6),(4,6),(3,6),(2,6),(1,6),(0,6)

TABLE A6: Routes and traffic demand for the *constant* traffic scenario. Each link in a route represents an intersection and the bottom left intersection in the grid-base city map is $(0, 0)$.

Glossary

Cycle is short for cycle length. It is the length of time required for all legal vehicle movements to occur.

Split is the proportion of green time given to a particular roadway within a cycle.

Offset is the time between the start of an upstream cycle and the start of a downstream cycle.

Platoon is a group of vehicles travelling in a 'convoy' along a roadway.

Cruise travel time is the estimated time required to travel from one intersection to another.

Phase is a portion of a traffic signal timing that is given to a set of vehicle movements.

Bibliography

[1] Robert L. Gordon and Warren Tighe. Traffic Control Systems Handbook. Technical Report FHWA-HOP-06-006, Dunn Engineering Associates, October 2005.

[2] Fanis Grammenos, Barry Craig, Douglas Pollard, and Carla Guerrera. Hippodamus Rides to Radburn: A New Model for the 21st Century. *Journal of Urban Design*, 13(2):163–176, June 2008.

[3] T. R. L. Limited. *SCOOT Advice Leaflet 1: The SCOOT urban traffic control system.* April 2016.

[4] Joe Hicks and Grahame Allen. *A Century of Change: Trends in UK statistics since 1900.* Social and General Statistics Section. House of Commons Library, 1999.

[5] Jeremy Grove. *Vehicle Licensing Statistics: Quarter 1 (Jan - Mar) 2016.* Department for Transport, 2016.

[6] Philip Gomm and Ivo Wengraf. The Car and the Commute, The journey to work in England and Wales. Technical report, The Royal Automobile Club Foundation for Motoring, December 2013.

[7] The economic costs of gridlock. Technical report, Centre for Economic and Business Research, December 2012.

[8] Panayotis Christidis and Juan Nicolas Ibanez Rivas. Measuring Road Congestion. Technical report, European Commission, Joint Research Centre, 2012.

[9] Stephen Knight, Murad Qureshi, James Cleverly, Len Duvall, Nicky Gavron, Jenny Jones, and Kit Malthouse. Driving Away from Diesel, Reducing Air Pollution from Diesel Vehicles. Technical report, London Assembly, Environment Committee, July 2015.

[10] Steven H. L. Yim and Steven R. H. Barrett. Public health impacts of combustion emissions in the United Kingdom. *Environmental Science & Technology*, 46(8): 4291–4296, March 2012.

[11] Frank J. Kelly and Julia C. Fussell. Air pollution and public health: emerging hazards and improved understanding of risk. *Environmental Geochemistry and Health*, 37(4):631–649, 2015.

[12] Sven Koenig, Craig A. Tovey, Michail G. Lagoudakis, Evangelos Markakis, David Kempe, Pinar Keskinocak, Anton J. Kleywegt, Adam Meyerson, and Sonal Jain. The Power of Sequential Single-Item Auctions for Agent Coordination. In *Proceedings of the 21st National Conference on Artificial Intelligence*, volume 2, pages 1625–1629. AAAI Press, 2006.

[13] Michael P. Wellman and Peter R. Wurman. Market-aware agents for a multiagent world. *Robotics and Autonomous Systems*, 24(3-4):115–125, 1998.

[14] Konstantinos Aboudolas, Markos Papageorgiou, and Elias Kosmatopoulos. Control and Optimization Methods for Traffic Signal Control in Large-scale Congested Urban Road Networks. *2007 American Control Conference*, 2007.

[15] Jeffery Raphael, Simon Maskell, and Elizabeth Sklar. From Goods to Traffic: First Steps Toward an Auction-based Traffic Signal Controller. In *13th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*, Cham, Switzerland, 2015. Springer International Publishing.

[16] Jeffery Raphael, Simon Maskell, and Elizabeth Sklar. An Empirical Investigation of Adaptive Traffic Control Parameters. In *Proceedings of the Workshop on Agents in Traffic and Transportation at IJCAI 2016*. CEUR-WS.org, 2016.

[17] Jeffery Raphael, Elizabeth I. Sklar, and Simon Maskell. An Intersection-centric Auction-based Traffic Signal Control Framework. In Amparo Alonso-Betanzos, Noelia Sanchez-Marono, Oscar Fontenla-Romero, Gary J. Polhill, Tony Craig, Javier Bajo, and Juan Manuel Corchado, editors, *Agent-Based Modeling of Sustainable Behaviors*, pages 121–142. Springer International Publishing, 2017.

[18] Roger P. Roess, Elena Prasas, and William R. McShane. *Traffic Engineering: International Edition, 4th Edition.* Pearson Education, Inc., 4th edition, 2009. ISBN 978-93-325-0936-8.

[19] Ana L. C. Bazzan. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, 18: 342–375, 2008.

[20] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Yibing Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, December 2003.

[21] James Bonneson, Srinivasa Sunkari, and Michael Pratt. Traffic Signal Operations Handbook. Technical report, March 2009.

[22] Transportation Research Board. *Highway Capacity Manual.* Transportation Research Board, National Research Council, Washington, D.C., 2000.

[23] Javed Alam and Manoj K. Pandey. Design and Analysis of a Two Stage Traffic Light System Using Fuzzy Logic. *Journal of Information Technology & Software Engineering*, 5(3), 2015.

[24] Jelka Stevanovic, Aleksandar Stevanovic, Peter T. Martin, and Thomas Bauer. Stochastic Optimization of Traffic Control and Transit Priority Settings in VISSIM. *Transportation Research Part C: Emerging Technologies*, 16(3):332–349, 2008.

[25] Bhargava Rama Chilukuri, Joseph Perrin, and Peter T. Martin. SCOOT and Incidents: Performance Evaluation in Simulated Environment. *Transportation Research Record: Journal of the Transportation Research Board*, 1867:224–232, 2004.

[26] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton. SCOOT - A traffic responsive method of coordinating signals. Technical Report 1014, Transport and Road Research Laboratory, 1981.

[27] Pitu Mirchandani and Fei-Yue Wang. RHODES to Intelligent Transportation Systems. *IEEE Intelligent Systems*, 20(1):10–15, 2005. ISSN 1541-1672.

[28] Nathan H. Gartner, Farhad J. Pooran, and Christina M. Andrews. Implementation of the OPAC Adaptive Control Strategy in a Traffic Signal Network. In *Proceedings of IEEE Intelligent Transportation Systems Conference*, Oakland, CA, USA, August 2001. IEEE.

[29] Nathan Gartner. OPAC: A demand–responsive strategy for traffic signal control. *Transportation Research Record*, 906:75–81, 1983.

[30] Dietrich Braess, Anna Nagurney, and Tina Wakolbinger. On a Paradox of Traffic Planning. *Transportation Science*, 39(4):446–450, 2005.

[31] Bo Chen and Harry H. Cheng. A Review of the Applications of Agent Technology in Traffic and Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):485–497, June 2010. ISSN 1524-9050.

[32] Ana L. C. Bazzan, Denise de Oliveira, and Bruno C. da Silva. Learning in Groups of Traffic Signals. *Engineering Applications of Artificial Intelligence*, 23(4):560–568, June 2010. ISSN 0952-1976.

[33] Yang Xu, Yulin Zhang, and Ming Liu. Multiagent Based Decentralized Traffic Light Control for Large Urban Transportation System. *Mathematical Problems in Engineering*, 2014, 2014.

[34] Ivo J. P. M. Timóteo, Miguel R. Araujo, Rosaldo J. F. Rossetti, and Eugénio C. Oliveira. Using TraSMAPI for the assessment of multi-agent traffic management solutions. *Progress in AI*, 1(2):157–164, 2012.

[35] Lynne E. Parker. Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems. *AAAI Fall Symposium*, 2007.

[36] Fei-Yue Wang. Agent-Based Control for Networked Traffic Management Systems. *IEEE Intelligent Systems*, 20(5):92–96, 2005.

[37] K. Tumer and A. Agogino. Distributed Agent-Based Air Traffic Flow Management. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 330–337, May 2007.

[38] Bo Chen, Harry H. Cheng, and Joe Palen. Integrating Mobile Agent Technology with Multi-Agent Systems for Distributed Traffic Detection and Management Systems. *Transportation Research Part C: Emerging Technologies*, 17(1):1–10, 2009.

[39] Reza Zanjirani Farahani, Elnaz Miandoabchi, W.Y. Szeto, and Hannaneh Rashidi. A review of urban transportation network design problems. *European Journal of Operational Research*, 229(2):281–302, 2013.

[40] Gerhard Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-23203-0.

[41] Ana L. C. Bazzan and Franziska Klugl. *Multi-Agent Systems for Traffic and Transportation Engineering*. IGI Global, Hershey, PA, 2009. ISBN 1-60566-226-7 978-1-60566-226-8.

[42] Birgit Burmeister, Afsaneh Haddadi, and Guido Matylis. Application of multi-agent systems in traffic and transportation. *Software Engineering - IEE Proceedings*, 144(1):51–60, 1997.

[43] Lior Kuyer, Shimon Whiteson, Bram Bakker, and Nikos A. Vlassis. Multiagent Reinforcement Learning for Urban Traffic Control Using Coordination Graphs. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *ECML/PKDD (1)*, volume 5211 of *Lecture Notes in Computer Science*, pages 656–671. Springer, August 2008. ISBN 978-3-540-87478-2.

[44] Stephen Chiu. Adaptive Traffic Signal Control Using Fuzzy Logic. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 98–107, 1992.

[45] James C. Spall and Daniel C. Chin. Traffic-Responsive Signal Timing for System-wide Traffic Control. *Transportation Research Part C: Emerging Technologies*, 5 (3):153–163, 1997.

[46] S. Araghi, A. Khosravi, and D. Creighton. Intelligent cuckoo search optimized traffic signal controllers for multi-intersection network. *Expert Systems with Applications*, 42(9):4422–4431, 2015.

[47] Bram Bakker, Shimon Whiteson, Leon J. H. M. Kester, and Frans C. A. Groen. Traffic Light Control by Multiagent Reinforcement Learning Systems. In Robert Babuska and Frans C. A. Groen, editors, *Interactive Collaborative Information Systems*, volume 281 of *Studies in Computational Intelligence*, pages 475–510. Springer, 2010. ISBN 978-3-642-11687-2.

[48] M.M. Abdelhameed, M. Abdelaziz, S. Hammad, and O.M. Shehata. A Hybrid Fuzzy-Genetic Controller for a multi-agent intersection control system. In *Second International Conference on Engineering and Technology*, 2015.

[49] Road safety in the European Union. Technical report, European Commission, Mobility and Transport DG, March 2015.

[50] Traffic Safety Facts Research Note. Technical report, National Highway Traffic Safety Administration, December 2014.

[51] Kurt M. Dresner and Peter Stone. Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism. In *Proceedings of the Third International Joint Conference on AAMAS*, pages 530–537. IEEE Computer Society, 2004. ISBN 1-58113-864-4.

[52] Kurt M. Dresner and Peter Stone. Multiagent traffic management: An improved intersection control mechanism. In Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael Wooldridge, editors, *AAMAS*, pages 471–477. ACM, 2005. ISBN 1-59593-094-9.

[53] Kurt M. Dresner and Peter Stone. A Multiagent Approach to Autonomous Intersection Management. *Journal Artificial Intelligence Research (JAIR)*, 31:591–656, March 2008.

[54] Kurt M. Dresner and Peter Stone. Multiagent Traffic Management: Opportunities for Multiagent Learning. In Karl Tuyls, Pieter Jan't Hoen, Katja Verbeeck, and Sandip Sen, editors, *LAMAS*, volume 3898 of *Lecture Notes in Computer Science*, pages 129–138. Springer, 2006. ISBN 3-540-33053-4.

[55] Matthew J. Hausknecht, Tsz-Chiu Au, and Peter Stone. Autonomous Intersection Management: Multi-intersection optimization. In *IROS*, pages 4581–4586. IEEE, 2011. ISBN 978-1-61284-454-1.

[56] M.M. Abdelhameed, M. Abdelaziz, S. Hammad, and O.M. Shehata. Development and evaluation of a multi-agent autonomous vehicles intersection control system. In *ICET 2014 - 2nd International Conference on Engineering and Technology*, 2015.

[57] Lee Gomes. *Hidden Obstacles For Google's Self-Driving Cars: Impressive Progress Hides Major Limitations Of Google's Quest For Automated Driving.* MIT Technological Review, (www.technologyreview.com), 2014.

[58] Todd Litman. Autonomous Vehicle Implementation Predictions. Technical report, Victoria Transport Policy Institute, December 2015.

[59] Ravi Shanker, Adam Jonas, Scott Devitt, Katy Huberty, Simon Flannery, William Greene, Benjamin Swinburne, Gregory Locraft, Adam Wood, Keith Weiss, Joseph Moore, Andrew Schenker, Paresh Jain, Yejay Ying, Shinji Kakiuchi, Ryosuke Hoshino, and Andrew Humphrey. Autonomous Cars, Self-Driving the New Auto Industry Paradigm. Technical report, Morgan Stanley Research, November 2013.

[60] Bill Boudette. Self-Driving Tesla Was Involved in Fatal Crash, U.S. Says. *The New York Times*, July 2016. https://www.nytimes.com/2016/07/01/business/self-driving-tesla-fatal-crash-investigation.html.

[61] M. Bernardine Dias, Robert Zlot, Nidhi Kalra, and Anthony Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7): 1257–1270, 2006.

[62] Simon Parsons, Juan A. Rodriguez-Aguilar, and Mark Klein. Auctions and Bidding: A Guide for Computer Scientists. *ACM Comput. Surv.*, 43(2):10:1–10:59, February 2011. ISSN 0360-0300.

[63] Isaac K. Isukapati and George F. List. Agent Based Framework for Modeling Operations at Isolated Signalized Intersections. In *Proceedings of the 18th IEEE International Conference on Intelligent Transportation Systems*, pages 2900–2906. IEEE, September 2015.

[64] Dustin Carlino, Stephen D. Boyles, and Peter Stone. Auction-based autonomous intersection management. In *Proceedings of the 16th IEEE Intelligent Transportation Systems Conference (ITSC)*, 2013.

[65] Matteo Vasirani and Sascha Ossowski. A computational market for distributed control of urban road traffic systems. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):313–321, 2011.

[66] Matteo Vasirani and Sascha Ossowski. A Market-Inspired Approach for Intersection Management in Urban Road Traffic Networks. *Journal Artificial Intelligence Research*, 43(1):621–659, January 2012. ISSN 1076-9757.

[67] Heiko Schepperle and Klemens Böhm. Auction-Based Traffic Management: Towards Effective Concurrent Utilization of Road Intersections. In *10th IEEE International Conference on E-Commerce Technology (CEC) and 5th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (EEE)*, pages 105–112. IEEE, 2008.

[68] Heiko Schepperle and Klemens Böhm. Agent-Based Traffic Control Using Auctions. In Matthias Klusch, Koen V. Hindriks, Mike P. Papazoglou, and Leon Sterling, editors, *Proceedings of the 11th International Workshop on Cooperative Information Agents (CIA)*, volume 4676, pages 119–133. Springer Berlin Heidelberg, 2007.

[69] Sebastian Iwanowski. Auction-based Traffic Control on Roads. In *Proceedings of the 7th World Congress on Intelligent Systems*, 2000.

[70] S. Iwanowski, W. Spering, and W.J. Coughlin. Road traffic coordination by electronic trading. *Transportation Research Part C: Emerging Technologies*, 11(5): 405–422, 2003.

[71] Mehdi Mashayekhi and George List. A Multi-agent Auction-based Approach for Modeling of Signalized Intersections. In *Second Workshop on Synergies Between Multiagent Systems, Machine Learning and Complex Systems*, Buenos Aires, Argentina, July 2015.

[72] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 2nd edition, 2012.

[73] Bruno Castro da Silva, Robert Junges, Denise de Oliveira, and Ana L. C. Bazzan. ITSUMO: An Intelligent Transportation System for Urban Mobility. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1471–1472. ACM, 2006. ISBN 1-59593-303-4.

[74] Marco Wiering. Multi-Agent Reinforcement Learning for Traffic Light Control. In *Proceedings of the 17th International Conference on Machine Learning*, pages 1151–1158. Morgan Kaufmann, San Francisco, CA, 2000.

[75] Marco Wiering, Jilles Vreeken, Jelle Van Veenen, and Arne Koopman. Simulation and optimization of traffic in a city. In *Intelligent Vehicles Symposium*, pages 453–458. IEEE, 2004.

[76] Samah El-Tantawy, Baher Abdulhai, and Hossam Abdelgawad. Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto. *IEEE Transactions on Intelligent Transportation Systems*, 14(3): 1140–1150, 2013.

[77] Prabuchandran K. J, Hemanth Kumar A. N, and Shalabh Bhatnagar. Decentralized learning for traffic signal control. In *7th International Conference on Communication Systems and Networks*, pages 1–6, 2015.

[78] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2):235–256, May 2002.

[79] S. Richter. Learning traffic control-towards practical traffic control using policy gradients. Technical report, Albert-Ludwigs-Universitat Freiburg, 2006.

[80] Mohamed A. Khamis and Walid Gomaa. Adaptive Multi-objective Reinforcement Learning with Hybrid Exploration for Traffic Signal Control Based on Cooperative Multi-agent Framework. *Engineering Applications of Artificial Intelligence*, 29: 134–151, March 2014. ISSN 0952-1976.

[81] Seung-Bae Cools, Carlos Gershenson, and Bart D'Hooghe. Self-Organizing Traffic Lights: A Realistic Simulation. In Mikhail Prokopenko, editor, *Advances in Applied Self-Organizing Systems*, pages 45–55. Springer London, 2013.

[82] Matteo Vasirani and Sascha Ossowski. Learning and Coordination for Autonomous Intersection Control. *Applied Artificial Intelligence*, 25(3):193–216, 2011.

[83] D. H. Wolpert and K. Tumer. Optimal Payoff Functions for Members of Collectives. *Advances in Complex Systems*, 4(2/3):265–279, 2001.

[84] Mark D Foy, Rahim F Benekohal, and David E Goldberg. Signal Timing Determination Using Genetic Algorithms. *Transportation Research Record*, 1365:108–115, 1992.

[85] Frederick Hayes-Roth. Rule-based Systems. *Communications of the ACM*, 28(9): 921–932, September 1985. ISSN 0001-0782.

[86] Ana L. C. Bazzan. A Distributed Approach for Coordination of Traffic Signal Agents. *Autonomous Agents and Multi-Agent Systems*, 10(2):131–164, 2005.

[87] Jose Cuena, G. Ambrosino, and M. Boero. A General Knowledge-Based Architecture for Traffic Control: The KITS Approach. *International Conference on Artificial Intelligence Applications in Transportation*, pages 153–171, 1992.

[88] Nicholas V. Findler and John Stapp. A Distributed Approach to Optimized Control of Street Traffic Signals. *Journal of Transportation Engineering*, 118:99–110, 1992.

[89] B. Foraste and G. Scemama. Surveillance and Congested Traffic Control in Paris by Expert System. *International Conference on Road Traffic Control*, pages 91–94, 1986.

[90] Bernd Wild. SAPPORO, Towards an Intelligent Integrated Traffic Management System. *International Conference on Artificial Intelligence Applications in Transportation Engineering*, pages 19–39, 1992.

[91] Emmanuel K. Dinanga and Marcia Pasin. Toward Equitable Vehicle-based Intersection Control in Transportation Networks. In F. Klügl, J. Vokrinek, and G. Vizzari, editors, *Proceedings of the 8th International Workshop on Agents in Traffic and Transportation*, 2014.

[92] Francois Dion and Bruce Hellinga. A rule-based real-time traffic responsive signal control system with transit priority: application to an isolated intersection. *Transportation Research Part B: Methodological*, 36(4):325–343, 2002.

[93] Sam Yagar and Bin Han. A procedure for real-time signal control that considers transit interference and priority. *Transportation Research Part B: Methodological*, 28(4):315–331, 1994.

[94] Stefan Lämmer, Dirk Helbing, and Dirk Helbing. Self-Stabilizing Decentralized Signal Control of Realistic, Saturated Network Traffic. *Transportation Science*, pages 484–496, 2006.

[95] Isabel Martí, Vicente R. Tomás, Arturo Sáez Esteve, and Juan J. Martínez. A Rule-Based Multi-agent System for Road Traffic Management. In *Web Intelligence/IAT Workshops*, pages 595–598. IEEE, 2009.

[96] Vicente Milanés, Joshué Pérez, Enrique Onieva, and Carlos González. Controller for Urban Intersections Based on Wireless Communications and Fuzzy Logic. *IEEE Trans. Intelligent Transportation Systems*, 11(1):243–248, 2010.

[97] Abdollah A. Shahraki, Meisam N. Shahraki, and Mohammad R. Mosavi. Design and simulation of a fuzzy controller for a busy intersection. In *International Conference on Computer Applications Technology*, pages 1–6. IEEE, January 2013. ISBN 978-1-4673-5284-0.

[98] Kok Khiang Tan, Marzuki Khalid, and Rubiyah Yusof. Intelligent traffic lights control by fuzzy logic. *Malaysian Journal of Computer Science*, 9(2):29–35, 1996. ISSN 0127-9084.

[99] P. G. Balaji, G. Sachdeva, Dipti Srinivasan, and Chen-Khong Tham. Multi-agent System based Urban Traffic Management. In *IEEE Congress on Evolutionary Computation*, pages 1740–1747. IEEE, 2007.

[100] Halim Ceylan and Michael G. H. Bell. Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing. *Transportation Research Part B: Methodological*, 38(4):329–342, 2004.

[101] Dennis Robertson. Research on the TRANSYT and SCOOT Methods of Signal Coordination. *ITE journal*, 56(1):36–40, 1986.

[102] Luis Miramontes Hercog. *Co-evolutionary Agent Self-Organization for City Traffic Congestion Modeling*, volume 3103 of *Lecture Notes in Computer Science*. Springer, 2004.

[103] Min Chee Choy, Dipti Srinivasan, and Ruey Long Cheu. Cooperative, Hybrid Agent Architecture for Real-Time Traffic Signal Control. *IEEE Transactions on*

Systems, Man, and Cybernetics - Part A: Systems and Humans, 33(5):597–607, 2003.

[104] Guillaume Leduc. Road Traffic Data: Collection Methods and Applications. Technical report, European Commission, Joint Research Centre, 2008.

[105] Jeffery Raphael, Eric Schneider, Simon Parsons, and Elizabeth I. Sklar. Learning a policy for collision avoidance by mining interactive multi-robot games. In *Workshop on Autonomous Robots and Multirobot Systems (ARMS) at Autonomous Agents and MultiAgent Systems (AAMAS)*, Paris, France, May 2014.

[106] Elizabeth Sklar, Simon Parsons, A Tuna Ozgelen, MQ Azhar, Todd Flyr, Eric Schneider, and Jeffery Raphael. A Practical Approach to Human/Multi-Robot Teams. In *11th European Workshop on Multi-Agent Systems (EUMAS)*, 2013.

[107] Ali Ekici, Pinar Keskinocak, and Sven Koenig. Multi-robot routing with linear decreasing rewards over time. In *ICRA*, pages 958–963. IEEE, 2009.

[108] T. Arai, E. Pagello, and L. Parker. Editorial: Advances in multi-robot systems. *IEEE Transactions on Robotics and Automation*, 18(5):655–661, 2002.

[109] Justin Melvin, Pinar Keskinocak, Sven Koenig, Craig A. Tovey, and Banu Yuksel Ozkaya. Multi-robot routing with rewards and disjoint time windows. In *IROS*, pages 2332–2337. IEEE, 2007.

[110] Vanessa Frías-Martínez, Elizabeth Sklar, and Simon Parsons. Exploring Auction Mechanisms for Role Assignment in Teams of Autonomous Robots. In Daniele Nardi, Martin A. Riedmiller, Claude Sammut, and José Santos-Victor, editors, *RobuCup*, volume 3276 of *Lecture Notes in Computer Science*, pages 532–539. Springer, 2004.

[111] Sven Koenig, Pinar Keskinocak, and Craig A. Tovey. Progress on Agent Coordination with Cooperative Auctions. In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010.

[112] Lawrence Klein, Milton K. Mills, and David R.P. Gibson. Traffic Detector Handbook: Third Edition—Volume I. Technical Report FHWA-HRT-06-108, U.S. Department of Transportation, Federal Highway Adminstration, October 2006.

[113] Martin Treiber and Arne Kesting. *Traffic Flow Dynamics*. Springer-Verlag Berlin Heidelberg, 2013.

[114] SCOOT User Guide. Technical Report 32, Siemens Mobility, Traffic Solutions, 1999.

[115] Sang Soo Lee, Seung Hwan Lee, Young Tae Oh, and Kee Choo Choi. Development of Degree of Saturation Estimation Models for Adaptive Signal Systems. *RSCE Journal of Civil Engineering*, 6(3):337–345, 2002.

[116] Luke Riley, Katie Atkinson, Paul E. Dunne, and Terry R. Payne. Distributing coalition value calculations to coalition members. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2117–2123, 2015.

[117] Tuomas Sandholm, Kate Larson, Martin Andersson, Onn Shehory, and Fernando Tohme. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, January 1999.

[118] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent Development and Applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[119] Stefan Krauss. Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics. *Deutsches Zentrum fuer Luft- und Raumfahrt. Forschungsberichte*, 1998.

[120] Dustin Carlino, Mike Depinet, Piyush Khandelwal, and Peter Stone. Approximately Orchestrated Routing and Transportation Analyzer: Large-scale Traffic Simulation for Autonomous Vehicles. In *Proceedings of the 15th IEEE Intelligent Transportation Systems Conference (ITSC)*, September 2012.

[121] R. D. Bretherton and G. T. Bowen. Recent Enhancements to SCOOT (SCOOT Version 2.4). *Third International Conference on Road Traffic Control*, pages 95–98, May 1990.

[122] Blake G. Hansen, Peter T. Martin, and H. Joseph Perrin. SCOOT Real-Time Adaptive Control in a CORSIM Simulation Environment. *Transportation Research Record*, 1727(00-1550):27–30, 2000.

[123] Dennis Robertson and R. David Bretherton. Optimizing Networks of Traffic Signals in Real Time-The SCOOT Method. *IEEE Transactions on Vehicular*, 40(1), February 1991.

[124] John A. Halkias. Demonstration Project No. 105: Advanced Transportation Management Technologies. Tech. Report FHWA-SA-97-058, United States. Federal Highway Administration, 1997.

[125] F. V. Webster. Traffic Signal Settings. Technical report, HMSO, London, 1958.

[126] H. B. Mann and D. R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50–60, March 1947.

[127] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. http://www.R-project.org.